Treball Final de Grau
d'Estadística Aplicada

Web Scraping de Vídeos de YouTube i Realització d'un Model Predictiu per a la seva Visualització

Nom: Arnau Rovira Chassignet
Tutor: Albert Ruiz Cirera
NIU: 1363022
Data: 4 de febrer de 2018

Universitat Autònoma de Barcelona

Facultat de Ciències
Curs 2017-2018

Índex

1 Introducció

L'objectiu d'aquest treball és la creació d'un model que em proporcioni la probabilitat de que jo visualitzi un vídeo concret de la pàgina web de visualització de vídeos online YouTube. Aquesta probabilitat ens permetrà fer una predicció per a cada vídeo, és a dir una variable que respongui a la pregunta "Veuré aquest vídeo?". Per a fer-ho usaré els últims vídeos que he visualitzat en el meu compte de YouTube i segons les variables del vídeo a evaluar se li adjudicarà una probabilitat o una altra.

He triat aquest tema per al meu treball final de grau perquè uso molt la web YouTube per a veure vídeos de les coses que m'interessen. Passo part del meu temps lliure veient vídeos i tinc curiositat per saber fins a quin punt se'm poden associar certs tipus de vídeo.

Actualment uso YouTube principalment per a escoltar música durant el meu dia a dia, però també com a suport mitjançant vídeos de tutorials, per a veure competicions de videojocs o fins i tot sèries o pel·lícules.

1.1 Història de YouTube

YouTube és una idea que sorgeix de tres amics: Chad Hurley, Steve Chen i Jawed Karim, que es van conèixer mentre treballaven en la mateixa empresa. Després d'assistir una festa on van realitzar diversos enregistraments es van trobar amb tot un seguit de problemes per a compartir aquests vídeos. Degut a aquesta necessitat van crear la plataforma web de visualització de vídeos YouTube. Mentre que Hurley i Chen defensen aquesta història, Karim diu que YouTube volia ser inicialment una web de cites on les persones es podrien calificar les unes a les altres a través dels seus vídeos.

Sigui com sigui, el 15 de febrer de l'any 2005, van crear la plataforma web YouTube. El primer vídeo va ser penjat pel cofundador Jawed Karim el 23 d'abril sota el nom "Me at the Zoo". Poc després els creadors van veure que els usuaris usaven la web per a penjar tot tipus de vídeos.

Durant l'any 2005 YouTube es va popularitzar molt, fins al punt d'aconseguir uns 50 milions de visites diàries cap a finals d'aquest mateix any. L'any 2006 l'empresa ja comptava amb aproximadament 60 treballadors i poc després que es publiqués l'adquisició de YouTube per part de Google, aquesta es va fer efectiva per uns 1.650 milions de dòlars. En el moment de la transacció es visualitzaven uns 100 milions de vídeos i s'hi afegien 65.000 vídeos nous al dia. Aproximadament 72 milions de persones la visitaven cada mes.

Cap a finals de la primera dàcada dels 2000 YouTube era ja la plataforma web líder en la visualització de vídeos. Des d'aleshores Google no va obtenir any rere any els beneficis esperats, arribant a ser considerada una web rentable.

Tot i això YouTube, especialment els últims anys, ha creat al seu voltant tot un fenòmen de masses. Aquest ha estat aprofitat per molts usuaris per a fer-se populars i poder arribar a guanyar-se la vida mitjançant els anuncis. A aquest seguit de persones se'ls anomena com a "youtubers".

Actualment YouTube és la tercera web més visitada del món, només per darrere de Google i Facebook i cada minut s'hi penjen 300 hores de contingut.

1.2 Estructura del Treball

Pel que fa a l'estructura d'aquest treball, el trobem dividit en diverses seccions. En la segona secció exposo el programari usat per a l'obtenció de les dades i les eines estadístiques aplicades. En el tercer apartat explico àmpliament el procés de programació dut a terme per a l'obtenció de les dades. En el proper apartat s'exposen les dades i les modificacions realitzades sobre aquestes. En la cinquena i sisena secció he aplicat

els models considerats a les dades i escollit quin dels models em semblava més adient per a la predicció de visualitzacions. Finalment exposo les conclusions extretes de la realització d'aquest treball. Als annexos hi podem trobar tots els gràfics i taules no inclosos en el cos del treball, així com el codi R usat.

2 Metodologia

Per a la realització d'aquest treball he fet servir un seguit de programes o softwares depenent de l'apartat o secció del treball i, per tant, de l'obectiu que volia aconseguir en aquell moment. Aquest treball es pot dividir en dos apartats principals, en primer lloc la recollida de dades i posteriorment el processament d'aquestes fins a aconseguir els models que em proporcionin les prediccions desitjades.

Pel que fa al procés de recollida de dades he decidit fer-lo amb codi Bash, és a dir en programari en base Linux. El motiu d'aquesta decisió ha estat la potència de programació i processament d'aquest sistema operatiu i posar en pràctica els aprenentatges establerts durant l'assignatura d'Eines Informàtiques i les assignatures de programació cursades durant la carrera.

Per tal de poder disposar d'aquest sistema operatiu sense haver de fer una gran inversió comprant un altre ordinador vaig optar en un primer moment per instalar una màquina virtual al meu sistema Windows. Vaig instalar el programari "Oracle VM VirtualBox" però em resultava molt incòmode i per recomanació del meu tutor, l'Albert Ruiz, vaig decidir canviar la manera d'accedir a Linux. Amb la seva ajuda vam fer una partició del disc dur del meu ordinador portàtil per tal poder tenir instalats tots dos sistemes operatius en un sol ordinador. El programari instalat és Ubuntu, un sistema operatiu en base Linux.

En el procés de recollida de dades es poden diferenciar fins a tres passos, cadascun amb un objectiu propi. En primer lloc hi trobem l'elaboració d'un seguit d'instruccions capaces d'extreure la informació necessària del codi de l'arxiu html corresponent a un vídeo qualsevol. Una vegada he confeccionat aquest codi el pas següent és elaborar, a partir del meu historial i d'una mostra aleatòria, un llistat de vídeos. Aquests seran els que conformaràn la base de dades. Finalment cal aplicar el codi inical a cadascun dels vídeos de la llista anterior i bolcar les dades en un fitxer per tal que posteriorment poguin ser analitzades.

Una vegada obtingudes les dades, per a l'anàlisi i la creació de models predictius he usat els coneixements i models estudiats i posats en pràctica en l'assignatura de Mineria de Dades. Durant l'estudi de les variables també he usat els coneixements adquirits en altres assignatures com per exemple en l'anàlisi descriptiu.

Durant aquesta segona part del treball diferencio fins a cinc passos abans d'obtenir el millor model predictiu. El primer pas és visualitzar les dades i fer estadística descriptiva per tal d'aplicar transformacions a les variables que així ho requereixin, juntament amb la creació de noves variables a partir de les dades originals. A continuació he hagut de decidir quines serien les variables definitives amb les quals crearia els diversos models. Com acabo de mencionar, un altre dels passos és la creació de diversos models i el càlcul d'estadistics varis. Finalment decidiré quin és el millor model predictiu a partir dels valors d'aquests estadístics calculats.

3 Recollida de Dades

Per a obtenir les dades del meu historial de la platarforma de visualització online de videos YouTube he usat la tècnica de "web scraping", que significa "escarbar una web". Aquesta tècnica consisteix en extreure informació de pàgines web de forma automatitzada. Les dades que es volen d'una web s'obtenen des del codi, generalment html, de la propia web en comptes de consultar-la. El més habitual és simular un ús humà d'internet mitjançant software d'ordinador.

En aquest cas el programa usat per a descarregar els fitxers html corresponents als vídeos del meu historial de YouTube ha estat la consola del sistema operatiu Ubuntu, que treballa en base Linux. El llenguatge estipulat d'ús en aquesta consola s'anomena Bash. Com he mencionat ja, he dicidit usar aquest programari per a posar en pràctica els coneixements adquirits durant la carrera i altres aventatges com és el seu ús lliure.

Per a poder realitzar els diversos models he pensat en un seguit de variables associades a cada vídeo. Aquestes són les següents:

- Durada del vídeo
- Canal que ha penjat el vídeo
- Subscripció del meu compte de YouTube al canal que ha penjat el vídeo
- Nombre de comptes de YouTube subscrits al canal que ha penjat el vídeo
- Nombre de visualitzacions
- Nombre de valoracions positives del vídeo
- Nombre de valoracions negatives del vídeo
- Nombre de comentaris dels usuaris
- Data de penjament del vídeo
- Categoria de YouTube en la que s'engloba el vídeo
- Vegades que s'ha compartit el vídeo
- Qualitat visual màxima del vídeo

El primer pas efectuat ha estat descarregar el fitxer .html corresponent a un vídeo qualsevol de YouTube per tal de veure quines de les variables que considerades en un principi era possible obtenir a partir de la informació continguda en el fitxer html. Mitjançant la funció "gawk", el funcionament de la qual veurem amb profunditat més endavant, amb la combinació d'expressions regulars he creat un seguit d'ordres. Aquestes ordres, una per a cada variable, aplicades al fitxer html corresponent al vídeo mencionat anteriorment retornen el valor de la variable en qüestió. Les variables amb les que he treballat finalment són la durada del vídeo, el nombre de comptes de YouTube subscrits a aquest canal, el nombre de visualitzacions, el nombre de valoracions positives del vídeo, el nombre de valoracions negatives del vídeo, la data de penjament del vídeo i la categoria de YouTube en la que s'engloba el vídeo.

Per a obtenir la base de dades cal tenir una sèrie d'arxius en un directori buit on es generaran alguns arxius a partir de l'execució de certes instruccions per la consola del sistema operatiu de Linux. Els arxius necessaris i les instruccions que s'han d'executar els veurem a continuació.

3.1 Historial de YouTube

Aquest ha de ser un arxiu amb extensió .html que contingui la informació corresponent a la llista de l'historial de visualitzacions d'un compte de Youtube. Com major sigui la llista de vídeos que contingui, millor ja que comptarem amb un major nombre de dades per poder entrenar el model que ens servirà per a fer la predicció de si veurem o no un vídeo qualsevol.

El fitxer ha estat extret mitjançant el navegador web Mozilla Firefox, accedint al compte de la plataforma de visualització de vídeos Youtube i carregant diverses vegades aquesta pestanya per tal d'obenir, com deiem abans, una gran quantitat de vídeos. Una vegada carregada la pàgina amb una bona cantitat de vídeos, mitjançant la drecera "ctrl+s" o a través del menú del navegador desem la pàgina en format .html.

El resultat d'aquesta acció és un fitxer amb una gran quantitat de codi entre el qual es troben els enllaços dels vídeos carregats en l'historial que posteriorment, mitjançant un seguit d'instruccions, recuperarem.

3.2 Fitxer vids1k

En aquest fitxer hi podem trobar 1000 identificadors de vídeos de youtube. Aquests han estat obtinguts a través de la web randomyoutube.net [3] que proporciona una mostra aleatòria de vídeos de YouTube. Tan sols hi apareixen 11 caràcters per fila, que són els corresponents a l'identificador de l'enllaç de cada vídeo. Aquest fitxer em servirà per a crear una base de dades de videos no vistos. Tot i que no puc estar segur que aquests vídeos no els hagi visualitzat mai, degut a la enorme quantitat de vídeos de la plataforma, la probabilitat que algun d'aquests vídeos hagi estat vist és pràcticament nul·la.

Aquesta mostra, però, no és del tot aleatòria. L'algoritme per aconseguir els vídeos consisteix en generar 11 caràcters aleatòris corresponents a l'identificador de l'enllaç d'un vídeo. D'aquest vídeo es busca el canal que l'ha penjat i es genera una llista de tots els vídeos penjats. S'elimina la primera entrada de la llista per tal d'obtenir vídeos antics i se'n selecciona un aleatòriament. Aquest procés es repeteix fins a obtenir els 1000 vídeos.

Aquest procés comporta certes limitacions, com per exemple que tots els vídeos penjats per un usuari que no ha penjat cap vídeo més queden fora de la mostra.

3.3 Fitxer instruccionsdefvistos

Aquest fitxer conté el codi Bash que emmagatzema les variables corresponents a un vídeo en paràmetres per finalment printar aquests paràmetres tot bolcant-los en un arxiu .csv. Per a obtenir els valors de les variables he usat la funció "gawk", que principalment consta de tres parts. A continuació podem veure un exemple de l'aplicació de la funció en una instrucció real:

```
agrada=$(gawk 'BEGIN{FS="[<>]"};/I like this/{print gensub(/,/,"", "g", $5)}' video.html)
```

La primera part, *gawk 'BEGIN{FS="[<>]"};* serveix per a definir quins seran els caràcters delimitadors de cada camp d'escriptura. En el llenguatge corresponent a la programació html la informació es distribueix en camps, on cada dada sol trobar-se separada de la resta en el seu propi camp. En aquest cas els delimitadors són els caràcters "[", "]", "<" i ">".

A continuació trobem la secció on s'especifica el text que serà buscat dins l'arxiu video.html. La secció de la instrucció és */I like this/*, i el codi buscat és el que es troba entre les dues barres "/", és a dir *I like this.*

Finalment tenim el següent codi: *{print gensub(/,/,"", "g", $5)}'*, que és on s'especifica el camp exacte que es vol seleccionar i quina acció se li vol aplicar. En aquest cas es el cinqué camp, i el que es vol es "imprimir-ho" o printar-ho. En aquest cas el resultat d'aplicar aquesta funció emmagatzema el nombre corresponent al total d'usuaris de YouTube als que els agrada el vídeo en la variable "agrada".

De la mateixa manera com hem vist en l'exemple he creat una instrucció per a cada variable, per tant en l'arxiu hi podem trobar un total de 7 instruccions i una més que podem veure a continuació:

```
printf "%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n" "$visites" "$agrada" "$noagrada" "$subscriptors" "$data" "$genere" "\"$durada\"" "1"
```

Aquesta instrucció serveix per a "imprimir" les variables especificades: *"$visites" "$agrada" "$noagrada" "$subscriptors" "$data" "$genere" "\"$durada\"" "1"*. Aquestes variables estaràn separades per tabuladors tal i com s'especifica amb el codi *"%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n"*. El grup de dades del vídeo quedarà separada per un salt de línea en respecte a les dades dels altres vídeos. Es pot observar però que hi ha un total de 8 variables i no 7 com comentava anteriorment. Això és degut a que la última variable, que sempre prendrà valor "1" per aquest arxiu executable correspón a si el vídeo ha estat visualitzat o no. Per aquest motiu aquest arxiu només s'aplicarà als vídeos obtinguts de l'historial i no als vídeos obtinguts del fitxer vids1k.

3.4 Fitxer instruccionsdefnovistos

Aquest fitxer té exactament la mateixa funció que el fitxer instruccionsdefvistos però com el seu nom indica per a vídeos no vistos. Per tant hi ha una única diferència entre ells, i és que el valor corresponent a la variable Visualitzat per a aquest fitxer serà sempre "0".

3.5 Fitxer instruccions

Aquest arxiu no té cap té cap tipus de sentit sense una petita explicació introductòria: durant la realització del treball, per tal de realitzar el codi més eficient, he estat descarregant vídeos per aplicar les instruccions i veure que funcionaven correctament. En un d'aquests casos em vaig trobar que les instruccions no extreien els valors corresponents i vaig descobrir que el codi html prenia un format diferent al que m'havia trobat fins al moment.

Durant la realització del treball YouTube va modificar el format del codi html, cosa que va suposar que hagués de crear de nou totes les instruccions que extreien la informació de cada vídeo adaptant-les a aquest nou format.

L'objectiu d'aquest fitxer és, per tant, poder obtenir la informació executant aquest arxiu, independentment del format html del vídeo i de si el vídeo procedeix de l'historial o de la base de dades de 1000 vídeos aleatoris de YouTube.

Per tal d'aconseguir-ho he usat la funció "if", que consisteix en que si una condició lògica proposada es compleix, s'executaràn una sèrie d'instruccions, en cas contrari, s'executen un altre seguit d'instruccions. A continuació podem veure les primeres línies del codi d'aquest arxiu:

```
visites=$(gawk 'BEGIN{FS="[<>]"};/watch-view-count/{print gensub(/,/,"", "g",gensub(/(.+) (.+)/, "\\1", "g", $15))}' video.html)
if [ "$visites"= ""]
then
```

```
visites=$(gawk 'BEGIN{FS="[<>]"};/<div id="count" class="style-scope ytd-video-primary-info-renderer"><yt-view-count-renderer class="style-scope ytd-video-primary-info-renderer"><span class="view-count style-scope yt-view-count-renderer"/print gensub(/,/,"", "g",gensub(/(.+) (.+)/, "\\1", "g", $7))' video.html)
...
else
agrada=$(gawk 'BEGIN{FS="[<>]"};/I like this/{print gensub(/,/,"", "g", $5)}' video.html)
...
fi
```

En primer lloc he definit la variable Visites amb un dels dos formats del codi html. A continuació amb el codi *if ["$visites"= ""]* uso la funció "if" i imposo com a condició lògica que la variable visites no contingui cap valor. En cas de ser així vol dir que l'arxiu conté l'altre format del codi html. En aquest cas s'aplicaríen les instruccions corresponents a l'altre format: *then visites=$(gawk 'BEGIN{FS="[<>]"};/<div id=çountçlass="style-scope ytd-video-primary-info-renderer"><yt-view-count-renderer class="style-scope ytd-video-primary-info-renderer"><span class="view-count style-scope yt-view-count-renderer"/print gensub(/,/,"", "g",gensub(/(.+) (.+)/, "\\1", "g", $7))' video.html) ....*

En cas de que la variable no estigués buida el codi que s'executaria correspondria a la resta d'instruccions per al mateix format html: *else agrada=$(gawk 'BEGIN{FS="[<>]"};/I like this/{print gensub(/,/,"", "g", $5)}' video.html) ....* D'aquesta manera en sortir del "if" ja tindria definides totes les variables menys la indicadora de si el vídeo ha estat visualitzat o no.

El següent pas és definir la variable Visualitzat. Per a fer-ho he requerit de dos "if" encadenats, de manera que si la primera condició no es compleix, passarà a al segon "if". Podem veure el codi a continuació:

```
link=$(gawk 'BEGIN{FS="[<>]"};/<meta property="og:url" content=/{print gensub(/,/,"", "g", $2)}' video.html | gawk 'BEGIN{FS="\""};//{print gensub(/,/,"", "g", $4)}')
if [ "$link"= ""]
then
visualitzat=""
else
aux=$(grep -c $link Historial.html)
if [ "$aux"= "0"]
then
visualitzat=0
else
visualitzat=1
fi
fi
```

En primer lloc, defineixo la variable link: *link=$(gawk 'BEGIN{FS="[<>]"};/<meta property="og:url" content=/{print gensub(/,/,"", "g", $2)}' video.html | gawk 'BEGIN{FS="\""};//{print gensub(/,/,"", "g", $4)}')*, que consisteix en un "gawk" que emmagatzema l'enllaç del vídeo d'on acabo d'extreure les dades. A continuació imposo com a condició lògica del primer "if" que la variable link no contingui cap valor (*if ["$link"= ""]*). En cas de ser així la variable Visualitzat no prendrà cap valor(*then visualitzat=""*), ja que vol dir que no s'ha pogut trobar el link del vídeo.

En cas de que el link hagi estat emmagatzemat es crea la variable aux. Per a definir-la he usat la funció "grep -c", que, donada una cadena de caràcters i un arxiu, efectua una búsqueda d'aquesta cadena dins l'arxiu i,

en cas de trobar coincidències, en retorna el nombre. En aquest cas la cadena serà el contingut de la variable link i l'arxiu, el corresponent a l'historial (*else aux=$(grep -c $link Historial.html*). En aquest moment entra en joc el segon "if", amb la condició lògica que aux sigui igual a zero (*if ["$aux"= "0"]*). Si es compleix aquesta condició vol dir que l'enllaç del vídeo no es troba contingut en el fitxer Historial i, per tant, que no ha estat visualitzat. Defineixo doncs la variable Visualitzat com a zero si és aquest el cas (*then Visualitzat=0*) i com a u en cas contrari (*else visualitzat=1 fi*).

Finalment, i com ja feia amb els arxius anteriors, l'últim pas és bolcar les variables definides en l'arxiu "fitxer.csv" tot separades per tabuladors. Per a fer-ho uso la instrucció "printf" que hem vist amb anterioritat.

D'aquesta manera obtinc un arxiu que en primer lloc evaluarà en quin format html està descarregat el vídeo, li aplicarà les instruccions corresponents i finalment "mirarà" si el vídeo ha provingut de l'historial o del fitxer vids1k i assignarà el valor corresponent a la variable Visualitzat.

3.6 Fitxer d'Instruccions

En aquest fitxer de text hi ha emmagatzemades les instruccions a executar per tal de poder obtenir, a partir dels fitxers anteriors, un fitxer .csv amb les dades de tots els vídeos de l'historial.

En primer lloc hi trobem la instrucció que ens permetrà extreure els enllaços de l'historial que abans he comentat i crearà un arxiu executable. La funció exacta d'aquesta instrucció és buscar un text especificat, que en ser trobat dins l'arxiu de l'historial, extraurà el camp especificat de la "frase" que el conté. Com podem veure a continuació la instrucció consisteix en un "gawk", funció de la qual ja he explicat el seu funcionament bàsic.

```
gawk 'BEGIN{FS="[<>]"};/yt-lockup-title contains-action-menu/{print gensub(/,/,"", "g", $8)}' Historial.html | gawk 'BEGIN{FS="\""};{print "&"$2}' | gawk 'BEGIN{FS="\&"};{print "lynx -source "$2" > video.html ; ./instruccionsdefvistos >> fitxer.csv"}END{print "rm video.html"}' > links
```

En aquest cas, però, la instrucció consta de diverses funcions "gawk" encadenades ja que en el camp especificat en la primera funció hi havia més text del que m'interessava. Per aplicar la funció gawk a la sortida de la funció anterior cal separar les fues funcions amb una barra vertical "|". En aquest cas els separadors dels camps seran les dues cometes " i simplement seleccionaré el camp desitjat, el segon. No tinc la necessitat de buscar cap text ja que la frase que m'interessa està ben delimitada.

Uso un altre cop la mateixa funcio gawk perque en alguns casos el link esta lligat al caràcter "&" i un text que no m'interessa, per tant uso aquest mateix caràcter com a delimitador. Com en el cas anterior no buscaré cap text, però a diferència del cas anterior no només printaré el camp selccionat. En aquest cas el que printaré serà el text següent juntament amb el segon camp, el seleccionat:

```
"lynx -source "$2" > video.html ; ./instruccionsdefvistos >> fitxer.csv"
```

El resultat és en sí mateix una instrucció, on el valor *$2* correspón al text que hi ha en el segon camp delimitat pel caràcter "&" que comentava abans, és a dir un enllaç d'un vídeo de YouTube. En bolcar aquest text en un arxiu de text buit es crearà un fitxer executable que anomenaré "links", que contindrà les instruccions per a extreure les variables per als vídeos vistos. El funcionament d'aquestes instruccions i resultat d'executar-les ho veurem una mica més endavant.

Com he comentat anteriorment durant el procés d'extracció de dades m'he trobat amb la problemàtica de l'existència de diferents formats html segons el moment en el que s'ha descarregat l'arxiu, o el navegador web usat per a fer-ho. Aquest fet afecta també, com vaig poder comprovar, a l'arxiu de l'historial, de manera que també patia canvis en el codi html. Per tant he creat una altra instrucció amb la mateixa funcionalitat per, en cas de desitjar-ho, poder executar algun altre historial i aplicar el que volia fer amb el meu compte a algun altre compte.

La següent instrucció d'aquest fitxer serveix per a crear l'arxiu "links2", que contindrà les instruccions per a obtenir les variables per als vídeos no vistos. Posteriorment veurem que aquest futur fitxer quedarà inclòs en l'arxiu "links", donat que ja no necessitaré un arxiu executable per a cada tipus de vídeo, sino que amb l'arxiu flexible instruccions serà suficient per als dos tipus de vídeo.

```
while read line; do printf "lynx -source https://www.youtube.com/watch?v=$line > video.html ; ./instruccionsdefnovistos >> fitxer.csv\n" >> links2; done < vids1k.txt
```

En aquest cas he usat un loop "while", *while read line; do ... done* de manera que mentre es detecti que hi ha una línia escrita en el fitxer vids1k, printi el text següent i el bolqui en l'arxiu "links2":

```
"lynx -source https://www.youtube.com/watch?v=$line > video.html ; ./instruccionsdefnovistos >> fitxer.csv\n"
```

Aquest text, com podem observar, és gairebé idèntic al que era bolcat en el fitxer "links". Canvia el fet que en aquest cas apaerix la part comuna dels enllaços dels vídeos YouTube ja que com he explicat en l'apartat corresponent al fitxer "vids1k", en l'arxiu només hi la part identificativa de l'enllaç. D'altra banda podem veure també que l'arxiu executable que s'aplicarà a aquests vídeos és el corresponent a vídeos no vistos.

Finalment afegeixo l'ordre d'eliminar l'arxiu video.html per tal de deixar el directori el màxim de polit possible.

El motiu pel qual he creat dos arxius amb els enllaços preparats per a ser descarregats i bolcats en el fitxer de dades final ha estat, com ja he comentat anteriorment, que alguns vídeos provenien de l'historial, i altres de la mostra aleatòria del fitxer vids1k. Per aquest motiu vaig crear dos arxius de lectura de fitxers, depenent de l'origen del vídeo. Aquest fet, però, ja no és necessari, ja que he creat el fitxer executable instruccions. Aquest té la capacitat de distingir si un vídeo pertany de l'historial o no, assignant d'aquesta manera el valor corresponent a la variable Visualitzat.

El text bolcat en els arxius links i links2 no és exactament el mostat en les caixes de codi corresponent a les funcions "lynx". Hi ha un petit canvi que consisteix en substituïr els fitxers instruccionsdefvistos i instruccionsdefnovistos pel fitxer instruccions, pel motiu que acabo d'explicar.

De la mateixa manera la instrucció corresponent a la funció "while" es veurà lleugerament modificada en substituïr el fitxer links2 pel links. Així totes les instruccions amb els enllaços per ser executades es trobaran en un sol arxiu.

La pròxima instrucció és per a crear l'arxiu de tipus .csv on hi haurà les dades finals. Per a crear l'arxiu el que fem és executar la següent instrucció:

```
printf "Visites\t Agrada\t NoAgrada\t Subscriptors\t Data\t Genere\t Durada\t Visualitzat\n" > fitxer.csv
```

Aquesta instrucció bolca a fitxer.csv el text especificat tot separat per tabuladors, d'aquesta manera en la primera fila del fitxer hi haurà el títol de cada variable. Tot i tenir una extensió .csv el fitxer és realment un .tsv en estar separat per tabuladors.

Les dues últimes instruccions serveixen per a executar els fitxers executables "links" i "links2" creats anteriorment i que a continuació veurem com funcionen.

```
sh links
```

```
sh links2
```

Els fitxers que hem vist fins ara són els necessaris inicialment per al procés d'obtenció de la base de dades. Com he explicat anteriorment per a la obtenció d'aquesta base de dades hem creat els fitxers "links" i "links2", que en ser executats bolcaràn les dades a fixer.csv. Passo a explicar per tant el contingut d'aquests dos fitxers, que és molt similar.

3.7 Fitxer links

Aquest és un fitxer que conté codi Bash i és el resultat d'haver executat la primera instrucció del fitxer d'instruccions, en ell hi tenim tantes instruccions com vídeos hi teníem en l'historial. En executar aquest fitxer aconseguirem les dades corresponents a aquests vídeos. Per a veure com es pot aconseguir això, veurem l'aspecte que presenten les instruccions que hi tenim en ell i el seu funcionament:

```
lynx -source https://www.youtube.com/watch?v=aLoGcf4lo1Y > video.html ; ./instruccionsdefvistos >> fitxer.csv
```

En aquest cas usarem la funció "lynx", que és un navegador web capaç de ser executat mitjançant una instrucció: *lynx -source*. El link *https://www.youtube.com/watch?v=aLoGcf4lo1Y*, anirà canviant per a cadascuna de les instruccions que podem trobar en aquest arxiu. Amb el codi > *video.html* el que he fet es bolcar el codi que conté la informació de la web en el fitxer video.html. Finalment amb *./instruccionsdefvistos >> fitxer.csv*, executo el fitxer "instruccionsdefvistos" i en bolco el seu resultat (els valors de cadascuna de les 8 variables), en el fitxer .csv creat anteriorment.

Totes les instruccions d'aquest arxiu són idèntiques exceptuant tan sols la secció identificativa de l'enllaç de YouTube. El procés es repetirà, per tant, tants cops com vídeos hi hagi en l'historial. Per a un total de 4231 enllaços, el temps d'execució usant una connexió de fibra ha estat de 50 minuts.

3.8 Fitxer links2

Aquest fitxer té la mateixa funció que el fitxer anterior però per als vídeos que no han estat vistos encara. Usaré exactament la mateixa instrucció amb un únic canvi, aquest consisteix en canviar el fitxer usat per a extreure les dades del fitxer video.html. En aquest cas el fitxer usat serà "instruccionsdefnovistos", que a diferència del "instruccionsdefvistos", el valor per a la variable visualitzat és 0 i no 1.

```
lynx -source https://www.youtube.com/watch?v=yEOqtAnjezw > video.html ; ./instruccionsdefnovistos >> fitxer.csv
```

Com en el cas anterior, d'una instrucció a una altra d'aquest fitxer, l'única diferència és l'identificador del vídeo. En aquest cas la llargada del fitxer és de 1000 instruccions, tantes com línies té el fitxer vids1k. En

aquest cas el temps d'execució ha estat de 15 minuts aproximadament.

Aquest arxiu, degut a les millores que he anat fent en el codi durant la elaboració del treball, en el procés final per a obtenir les dades no va arribar a ser creat, ja que queda englobat en "links", però he trobat adient exposar el seu contingut i funcions per tal d'entendre el procés de millora que he realitzat sobre el codi.

4 Tractament de les Dades

Per a cadascuna de les parts principals d'aquest treball he usat softwares diferents, més adients i còmodes per a mi segons la funció que hagués de cobrir. Fins ara el software usat era programari de Linux, la seva consola i l'ús d'expressions regulars. Una vegada obtingudes les dades, he decidit usar R mitjançant l'entorn R-Studio.

Tot i que hagués pogut usar el programa R en la línia d'instrucció del mateix sistema Linux, degut a que durant la meva trajectòria he estat usant el programari R-Studio, hi estic molt més acostumat i amb ell treballo molt més a gust.

Una vegada obtingut el fitxer .csv amb les dades de l'historial de YouTube el pas següent és depurar, analitzar i crear un seguit de models que em permetin predir la visualització d'un vídeo qualsevol de la plataforma YouTube, que és l'objectiu principal del treball. Tot el codi R usat en aquest apartat es pot trobar en l'Annex A.

4.1 Lectura i depuració de les dades

El primer pas consisteix en llegir les dades amb el programari R. Anomenaré a aquest conjunt inicial de les dades com a "dades". Una vegada carregades les dades el proper pas ha estat fer-ne una criba. He eliminat els registres repetits, deixant-ne un de sol, per tal de no tenir una base de dades redundant. Les dades que no s'han llegit correctament i per tant tenen algun valor que no correspon a la variable que els conté (text en una variable numèrica, per exemple) també han estat eliminades, així com les dades amb algun valor buit. A continuació he eliminat també les dades de les variables categòriques que corresponien a les categories amb menys de 10 observacions. El motiu d'aquest cribatge l'explicaré posteriorment. Les dades no aptes per a l'estudi i per tant eliminades corresponen aproximadament a un 15% del total de la mostra, de manera que seguim conservant una gran quantitat de dades. Defineixo acom a "dades2" aquest conjunt de dades depurat.

Com ja he avançat durant l'apartat d'obtenció de les dades, les variables extretes mitjançant el procediment de web scrapping finalment no han estat totes les que havia planejat inicialment. Això és degut a que en l'arxiu html no es troba tota la informació que jo esperava obtenir. Les variables que he pogut recollir són, per tant, les següents:

- **Visites**: variable numèrica que comptabilitza el nombre de visites del vídeo.
- **Agrada**: variable numèrica que comptabilitza el nombre d'usuaris que han especificat que el vídeo els agrada.
- **NoAgrada**: variable numèrica que comptabilitza el nombre d'usuaris que han especificat que el vídeo no els agrada.
- **Subscriptors**: variable de text que contè el nombre d'usuaris de YouTube subscrits al canal que ha penjat el vídeo en la plataforma. Les observacions amb un valor alt es mostren en milers o milons, per tant en alguns casos el valor no és exacte.
- **Data**: data en la que el canal va penjar el vídeo a la plataforma. El format utilitzat és "any-mes-dia".
- **Genere**: variable categòrica que indica la categoria de YouTube en la que es troba englobat el vídeo.
- **Durada**: variable de text que contè la durada del vídeo. El format utilitzat és "minuts:segons".
- **Visualitzat**: variable dicotòmica (0 o 1) que indica si el vídeo ha estat visualitzat a través del compte de Youtube que s'estudia o no. És la variable d'interés d'aquest treball.

4.2 Noves Variables

Les variables de l'apartat anterior són totes les obtingudes, però algunes d'aquestes es troben en un format que ens impedeix treure'n tota la informació que ens poden aportar. Per aquest motiu he decidit crear algunes variables noves, que substituiràn les variables que ens aporten menys informació. Aquestes noves variables són les següents:

- **Segons**: variable numèrica que conté la durada del vídeo en segons. Ha estat creada a partir de la variable Durada mitjançant la funció mmss_to_ss. Aquesta nova variable és numèrica, mentre que la anterior, en ser un text, era considerada categòrica (amb un gran nombre de categories).
- **DataCat**: variable categòrica que indica el període de temps en que va ser penjat el vídeo. Els períodes que es contemplen són 4, els anys 2005 a 2008, 2009 a 2011, 2012 a 2014 i 2015 a 2018. En aquest cas passem de tenir la variable Data amb centenars de categories (i per tant molt poques dades en cadascuna d'aquestes) a questa nova variable de quatre categories que resulta molt més útil.
- **Subscriptors**: variable numèrica que conté el nombre nombre d'usuaris de YouTube subscrits al canal que ha penjat el vídeo. És una variant de la variable Subcriptors, que s'ha obtingut aplicant la funció "transformador" (veure el codi R usat en l'AnnexA) a la variable que comentava. Els valors de la variable no són els valors exactes degut a que a les observacions amb un valor superior a 1000 queden arrrodonides a centenars, les superiors a 10000 a milers, i així successivament.

4.3 Anàlisi descriptiu de les variables

En primer lloc abans de començar a estudiar les variables cal definir-les a l'entorn del programari com el tipus de variable que els correspon. És a dir redefinir com a numèriques, categòriques, o factors totes aquelles variables que no tinguin un format adequat.

4.3.1 Anàlisi Univariant

En aquest apartat veurem les característiques de distribució de cadascuna de les variables amb les que vull treballar. Per a fer-ho visualitzaré els principals estadistics descriptius i realitzaré un gràfic de caixa.

Com deia, realitzo un petit resum de les dades per veure'n els estadístics bàsics com la mitjana, el màxim, el mínim i els quartils. Aquestes dades m'ajudaràn a veure si alguna variable és sospitosa de tenir una distribució extranya. Podem veure el resultats en la Taula 1.

	Visites	Agrada	NoAgrada	Subscriptors	Segons
Min.	1.000e+00	0	0	2	2
1r Qu.	1.074e+04	82	3	2100	190
Mediana	2.825e+05	2454	95	126000	280
Mitjana	1.843e+07	101533	4565	1121877	3288
3r Qu.	2.562e+06	21215	797	431000	635
Max.	2.567e+09	10454610	754127	25000000	7696730

Taula 1: Taula resum dels principals estadístics bàsics de les variables numèriques.

Es pot observar que totes les variables tenen una mitjana molt allunyada de la seva mediana. De fet en tots els casos la mitjana supera àmpliament al 3r quartil. Per a visualitzar millor les distribucions podem veure els primers diagrames de caixa de les Figures 1, 4, 5, 6 i 7.

D'altra banda pel que fa a les variables categòriques he realitzat una taula per a cada variable per tal de veure el nombre de categories que prenen i el nombre d'observacions que hi ha en cada categoria. Podem

veure les distibucions de freqüències per a les variables Genere, DataCat i Visualitzat en les Taules 7, 8 i 9 respectivament.

Es pot observar que per a la variable Genere hi ha categories amb una frequència inferior a 10, cosa que pot casuar problemes a l'hora de crear una base de dades d'entrenament i una de validació. Posteriorment veurem els motius i la transformació que realitzaré per evitar aquests problemes.

4.3.2 Transformació de Variables

Les dades exposades anteriorment, com hem vist, tenen una distribució molt poc agradable i per tant no han estat exactament les usades per a la realització dels models. Als casos mencionats els he realitzat una transformació. Aquests canvis els he dut a terme per tal de facilitar la manipulació de les dades així com per millorar les seves propietats. Les transformacions aplicades es poden veure a continuació.

Donades les distribucions que prenen les dades de les variables Segons, Agrada, NoAgrada i Subscriptors, totes molt similars, he cregut adient aplicar una transformació logarítmica. En un primer moment vaig aplicar la transformació $log(x)$, però les variables resultants eren inapropiades per a la realització de models. Això era degut a que les variables originals, tot i que en una baixa proporció, conten el valor 0, i per tant el valor resultant de la transformació era menys infinit. Una vegada identificat el problema he decidit variar lleugerament la transformació i aplicar la fórmula $log(x + 1)$.

Creo "dades3" a partir de la base de dades anterior. En aquesta nova base és on s'emmagatzemarán les noves variables obtingudes a partir de transformacions. Han resultat doncs les variables logSegons, logAgrada, logNoAgrada i logSubscriptors, que tot i no ser normals tenen una distribució molt més agradable i semblant a una normal que les variables anteriors. Podem observar les seves distribucions en contrast amb les de les variables sense aplicar la transformació en les Figures 1, 4, 5, 6 i 7.

La variable categòrica DataCat no ha patit cap modificació, mentre que per a la variable Genere he eliminat les categories amb una freqüència inferior a 10, aquestes són les categories "Trailers" i "Shows". Aquest pas, com deia ja anteriorment, és necessari per tal d'evitar problemes a l'hora de crear una base de dades d'entrenament i una de validació, ja que és necessari que hi hagi vídeos de totes les categories en ambdues bases de dades. Considero que 10 és una frequència suficientment gran com per que la probabilitat que passi el que he mencionat sigui fiable. En la Taula 2 podem veure la distribució definitiva de la variable Genere.

De les variables que tinc actualment n'he fet una selecció per a realitzar els diversos models. Degut a la distribució de les variables originals, les variables numèriques seleccionades han estat logAgrada, logNoAgrada, logVisites, logSubscriptors i logSegons, mentre que les variables categòriques han estat Genere i DataCat.

4.3.3 Anàlisi Bivariant

En aquest apartat estudiaré la relació entre les variables que he considerarat per a la creació dels models. Tant entre elles mateixes, com amb la variable objectiu del nostre model. Per a fer-ho primer m'he centrat en les variables numèriques i posteriorment en les categòriques.

En primer lloc he calculat la correlació que hi ha entre les variables numèriques per tal de veure la possible relació entre elles. En la Taula 3 hi podem trobar les seves correlacions.

El resultat és que sí estàn correlacionades, de fet ho estan molt. Les parelles de variables logAgrada-logNoAgrada, logAgrada-logVisites i logNoAgrada-logVisites tenen un coeficient de correlació superior a

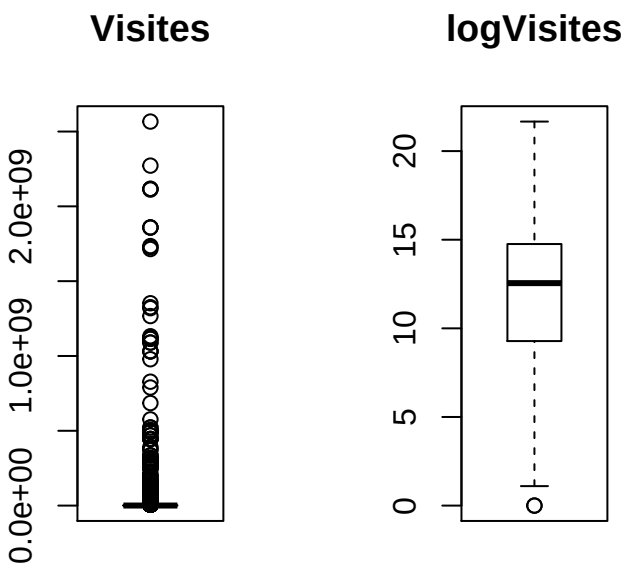

Figura 1: Diagrames de caixa per a les variables Visites i logVisites.

0.9 en tots 3 casos. Podem veure més clarament la relació entre aquestes parelles de variables en la Figura 8.

Es pot observar una marcada tendència lineal en els núvols de punts dibuixats per les parelles de variables comentades anteriorment. Aquest és un clar indicador que hi ha una alta colinealitat entre aquestes tres variables.

Això podria resultar un problema a l'hora d'explicar el funcionament del model ja que aquestes variables expliquen, en gran part, el mateix. Per tant, si es trobessin en un mateix model explicatiu seria contraproduent. En aquest cas com el que vull elaborar és un model predictiu no és un problema, ja que de com més informació disposi, millor serà l'estimació.

D'altra banda he decidit dividir la base de dades segons la variable resposta, és a dir, Visualitzat. Amb les dues bases de dades he decidit fer el mateix resum per a cadascuna de les variables. Per a les variables numèriques he decidit fer un reum dels principals estadístics bàsics. Aquest pas em servirà per veure si hi ha alguna variable que es comporti de manera diferent segons el valor de Visualitzat. En les Taules 4 i 5 podem trobar els valors d'aquests estadístics bàsics.

Com es pot observar, els estadístics per a les variables logVisites, logAgrada, logNoAgrada i logSubscriptors difereixen molt entre tots dos grups de dades. Per tal de visualitzar millor aquesta diferència he realitzat els seus diagrames de caixa en un sol gràfic per cada variable. Podem trobar-los en les Figures 2, 9, 10 i 11.

Veient els gràfics podem afirmar que les distribucions canvien molt depenent del valor que prèn la categoria Visualitzat. En casos com aquest podria ser que aquestes variables, en ser tan diferents, siguin decisives a

	Gènere
Autos & Vehicles	53
Comedy	135
Education	61
Entertainment	555
Film & Animation	114
Gaming	1089
Howto & Style	51
Music	1281
News & Politics	77
Nonprofits & Activism	13
People & Blogs	601
Pets & Animals	18
Science & Technology	93
Shows	0
Sports	162
Trailers	0
Travel & Events	34

Taula 2: Distribució definitiva de la variable Genere.

	logSegons	logVisites	logAgrada	logNoAgrada	logSubscriptors
logSegons	1.00	0.06	0.05	-0.01	0.27
logVisites	0.06	1.00	0.97	0.93	0.71
logAgrada	0.05	0.97	1.00	0.95	0.72
logNoAgrada	-0.01	0.93	0.95	1.00	0.65
logSubscriptors	0.27	0.71	0.72	0.65	1.00

Taula 3: Correlacions entre les variables numèriques.

l'hora de diferenciar entre vídeos vistos i no vistos.

Pel que fa a les variables categòriques he calculat i comparat les frequències relatives de les diferents categories segons si el vídeo ha estat visualitzat o no. En les Taules 10 i 11 podem trobar els valors obtinguts de les variables Genere i DataCat respectivament.

Hi ha certes diferències entre les categories de la variable Datacat, però no són tan notables com en la variable Genere. Podem veure que en les categories Music, Gaming i People & Blogs les diferències són clares, de fins a gairebé 40 punts. Per tant, depenent de la categoria del vídeo a classificar, aquesta variable pot jugar un paper important.

	logSegons	logVisites	logAgrada	logNoAgrada	logSubscriptors
Min.	1.099	2.773	0.000	0.000	1.099
1r Qu.	5.333	11.287	6.506	3.135	10.463
Mediana	5.680	13.455	8.802	5.472	12.101
Mitjana	6.071	13.242	8.486	5.262	11.729
3r Qu.	6.508	15.180	10.439	7.124	13.377
Max.	15.856	21.666	16.163	13.533	17.034

Taula 4: Taula resum dels principals estadístics bàsics de les variables numèriques dels vídeos visualitzats.

	logSegons	logVisites	logAgrada	logNoAgrada	logSubscriptors
Min.	1.386	0.000	0.0000	0.0000	1.099
1r Qu.	4.382	3.178	0.0000	0.0000	4.984
Mediana	5.347	4.394	0.6931	0.0000	6.038
Mitjana	5.296	4.874	1.3026	0.3657	5.899
3r Qu.	6.172	6.221	1.7918	0.0000	6.658
Max.	10.252	15.204	11.1957	6.9508	16.588

Taula 5: Taula resum dels principals estadístics bàsics de les variables numèriques dels vídeos no visualitzats..

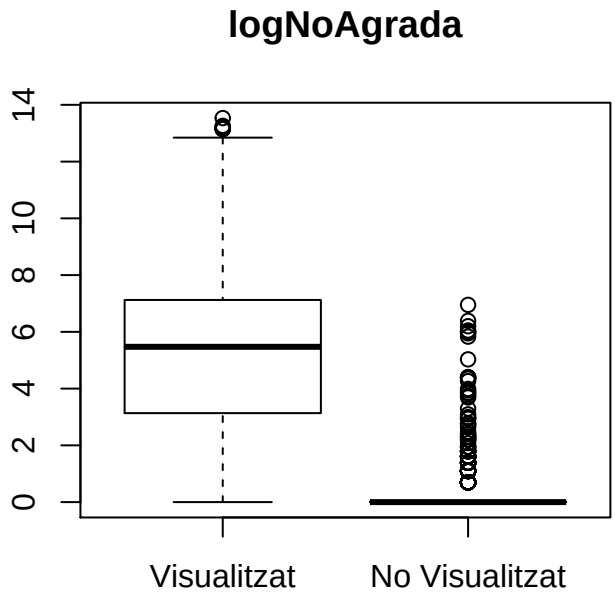

Figura 2: Diagrames de caixa per a la variable logNoAgrada segons el valor de la variable Visualitzat.

5 Creació de Models

Les variables que utilitzaré per a la creació dels models són les que es troben en la base de dades "dades4", i són les següents: Genere, Visualitzat, DataCat, logSegons, logVisites, logAgrada, logNoAgrada i logSubscriptors. Tot i que, com he pogut intuïr abans durant l'anàlisi bivariant, algunes d'aquestes variables ens aporten pràcticament la mateixa informació, he decidit mantenir-les dins els models donat que el meu objectiu és obtenir un model predictiu i no explicatiu.

Per tal de decidir quin dels models que crearé té una major potència predictiva, he considerat calcular diversos peràmetres. Aquests són l'error quaràtic mitjà (EQM), l'àrea sota la corba ROC (AUC) i el percentatge d'error de classificació.

Per a realitzar aquests càlculs, de la base de dades final de 4337 observacions completes, n'he creat dues: una d'entrenament amb aproximadament tres quartes parts de les observacions (al voltant de 3200 dades), que usaré per realitzar els models i una altra de validació amb el quart de les dades restants (unes 1100 dades). Amb els models realitzats amb base de dades d'entrenament realitzaré una predicció de la varible resposta per a les dades de la base de dades de validació. Contrastant els valors reals amb els predits per els models amb l'ajuda dels estadístics comentats anteriorment decidiré quin considero el millor model predictiu.

Per a cada model aplicat realitzaré els mateixos passos. En primer lloc crearé el model amb la base de dades d'entrenament. Posteriorment realitzo una predicció dels valors de la variable resposta per a les dades de la base de validació i finalment calcularé els valors dels estadístics que usaré per a l'elecció final del millor model predictiu. Per a cada mètode presentaré dos models, el primer model realitzat, i el millor model obtingut a partir de variar els paràmetres disponibles.

5.1 Model GLM

Els primers tipus de models usats seran els models lineals generalitzats o GLM. Com el seu nom indiquen són una generalització de la regressió lineal ordinària, que unifica diversos models estadístics tals com la regressió logística, de Poisson o la mateixa regressió lineal. Mitjançant la funció "link" o d'enllaç usada permeten modelar diversos tipus de variables com per exemple binomials, gaussianes o Poissons.

En aquest cas, donat que la variable resposta pren una distribució binomial, la funció d'enllaç serà la "logit". (Veure codi en l'Annex A.)

Per tal d'avaluar el model em basaré principalment en l'àrea sota la corba ROC i secundàriament en l'error quadràtic mitjà. En aquest model els valors obtinguts respectivament són 0.9844 (veure Gràfic 12) i és 0.0305.

5.2 Arbres de Classificació

El següent tipus de model usat són els arbres de clasificació. Aquests són una variant dels arbres de regressió per a variables resposta categòriques. El seu funcionament radica en realitzar tantes particions de les variables predictores com és possible i per a cadascuna d'aquestes en calcula un índex de puresa, concretament l'índex de Gini. S'escull la partició que minimitza aquest índex i a partir d'aquesta es torna a aplicar el mateix procediment múltiples vegades. Un paràmetre a tenir em compte és el paràmetre de complexitat de l'arbre, (cp).

Creo un arbre de classificació amb els paràmetres per defecte, és a dir un $cp = 0.01$. (Veure codi en l'Annex A.)

Els valors de l'àrea sota la corba ROC i l'error quadràtic mitjà obtinguts són 0.9399 i 0.0395, respectivament. Per a visualitzar a corba ROC i l'estructura d'aquest arbre veure els Gràfics 15 i 16.

Creo un altre arbre de regressió però aquest cop el valor del paràmetre *cp* serà de 0.001. (Veure codi en l'Annex A.)

En aquest cas els valors de l'àrea sota la corba ROC i l'error quadràtic mitjà són 0.9573 i 0.0308, respectivament. En el Gràfic 15 podem veure la corba ROC, i en el Gràfic 16 l'estructura de l'arbre.

5.3 Xarxes Neuronals

Les xarxes neuronals com el seu nom indiquen s'inspiren en les xarxes de connexions de les neurones del cervell. El funcionament de les xarxes neuronals es basa en la creació de combinacions lineals de les variables explicatives per a posteriorment modelar la resposta mitjançant funcions no necessàriament lineals d'aquestes combinacions. El nombre de capes ocultes i el nombre de nodes d'aquestes determinarà la complexitat de la xarxa. L'algoritme de maximització intern depén dels valors inicials dels pesos, que són aleatoris, per tant, l'aleatorietat hi juga un paper important. De fet executar la mateixa instrucció per a crear una xarxa neuronal dóna lloc a diferents xarxes neuronals.

Passo a crear una xarxa neuronal que serà executada tres cops per tal de que l'atzar no ens jugui una mala passada i, que tot i ser un bon model, obtinguem un mal resultat. Recordem que el resultat d'executar el codi d'una xarxa neuronal canvia cada cop. En aquesta primera xerxa neuronal m'he decidit per una capa oculta de 4 nodes i un màxim de 500 iteracions. (Veure codi en l'Annex A.)

Aquesta xarxa neuronal proporciona uns valors de l'àrea sota la corba ROC i l'error quadràtic mitjà de 0.9844 i 0.0305, respectivament. Podem veure la corba ROC en el Gràfic 17.

Proseguim amb una altra xarxa neuronal augmentant el nombre de nodes de la capa oculta fins a 6 i amb un màxim de 1000 iteracions. D'altra bada considero també la possibitat que les variables predictores es poguin relacionar directament amb la variable resposta sense la necessitat d'haver de passar per la capa oculta. (Veure codi en l'Annex A.)

Els resultats per aquesta xarxa més complexa que l'anterior són un AUC de 0.9699 i un EQM de 0.0345. Podem visualitzar la corba ROC en el gràfic 18.

5.4 Random Forests

L'últim mètode usat han estat els random forests. Els random forests consisteixen en l'elaboració de una gran quantitat d'arbres de regressió (B) amb una selecció aleatòria de les variables que disposem (m). Cada arbre serà realitzat a partir d'una mostra alatòria amb reposició de la mateixa mida que la base de dades original (n). Tindrem per tant un seguit de dades repetides en la base de dades usada per a crear l'arbre i algunes dades que no s'hauràn usat. Aquestes serviran per a calcular l'error de cada arbre i poder proporcionar un indicador de l'error global calculat a partir de tots els arbres creats.

Tot i ser un mètode amb un factor aleatòri considerable, en crear una gran mostra d'arbres on tant les variables com les dades són diferents, el que aconseguim és una gran varietat d'on usarem l'arbre més representatiu. Per aquest motiu, a diferència que amb les xarxes, no executo el model diversos cops.

En aquest primer arbre he considerat un nombre de 500 arbres on el mínim nombre de dades per a cada fulla és d'una observació. El nombre de variables aleatòriament escollides per a cada partició és de 3, donat pel valor per defecte. (Veure codi en l'Annex A.)

Per aquest primer model de random forest els valors de l'àrea sota la corba ROC i l'error quadràtic mitjà són 0.9888 i 0.0251, respectivament. Podem veure la corba ROC en el Gràfic 19.

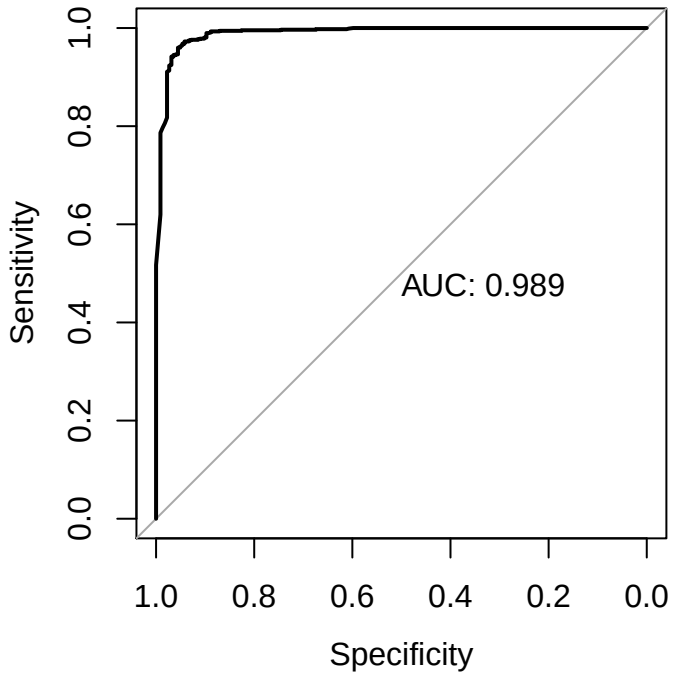

Figura 3: Corba ROC del random forest final.

En aquest segon model amplio el nombre d'arbres fins a 800 i redueixo el nombre de variables a 2 en cada partició. (Veure codi en l'Annex A.)

Els valors obtinguts en aquest cas són de 0.9892 i 0.0253 per a l'AUC i l'EQM respectivament i podem veure la corba ROC en el Gràfic 3.

6 Resultats

Com hem pogut veure tots els resultats corresponents a l'àrea sota la corba ROC són molt elevats. En cap dels estudis que he realitzat anteriorment havia obtingut uns valors tan alts, de manera que en un primer moment he dubtat d'estar realitzant correctament els models.

Tot i així, després d'executar els models diversos cops i comprovar que no hi havia cap error he recordat les grans diferències que presentaven les variables numèriques segons el valor de la variable resposta. Com he comentat durant l'anàlisi bivariant i he pogut comprovar afegint les variables d'una en una en els models, aquestes fan que les prediccions siguin molt precises.

En la Taula 6 podem veure els valors de l'AUC i l'EQM dels millors models per a cadascun dels mètodes usats.

	Model Lineal	A. Classificació	X. Neuronal	R. Forest
AUC	0.9844	0.9573	0.9844	0.9892
EQM	0.0305	0.0308	0.0305	0.0253

Taula 6: Taula dels AUC i EQM dels millors models de cada mètode per a les variables definitives.

Com deiem abans tots els mètodes obtenen una àra sota la corba ROC de pràcticament u, amb una diferència entre el millor i el pitjor model de 0.0319. Aquesta diferència és mínima en referència als valors de l'àrea sota la corba ROC obtinguts.

Pel que fa a l'error quadràtic mitjà tots els models prediuen molt bé les dades i els valors són en tots els casos gairebé zero.

6.1 Tria del millor model

A partir dels resultats que acabem de veure m'he decidit pel model obtingut amb el mètode del random forest. Aquesta decisió la prenc basant-me en els valors de l'àrea sota la corba ROC i l'error quadràtic mitjà. En tots dos casos els valors són els més favorables.

Tot i això, tots els mètodes obtenen valors molt similars per aquests estadístics, de manera que qualsevol dels mètodes seria una bona elecció per a predir la visualització de vídeos.

7 Conclusions

Les principals conclusions que he pogut extreure d'aquest estudi han estat que els tipus de vídeo que acostumo a veure a YouTube no són en cap cas representatius del contingut de la web.

Això ho hem pogut intuïr ja en l'anàlisi de les dades, cosa que en realitzar els models ha quedat confirmada. Les dades dels vídeos visualitzats per mi són tan diferents als que conformen la totalitat de la web que tots i cadascun dels mètodes usats han predit amb un error gairebé mínim els vídeos de la base de dades de validació. Podem veure molt clarament aquest fet en els Gràfics 13 i 14, on amb una sola divisió d'una variable obtenim fins a un àrea la corba ROC de 0.9399.

D'altra banda els vídeos corresponents a l'historial estan totalment lligats a les últimes tendències personals. Per exemple, podria passar que últimament hagi estat usant Youtube per a escoltar música i en el moment d'extreure l'historial hi ha un major nombre de vídeos musicals que els que visualitzo normalment. Això és així degut a que no he trobat la manera obtenir el meu historial complert.

D'altra banda, pel que fa a la secció d'extracció de dades, he pogut comprovar que com més curt, simple i versàtil és un codi, més profit se'n pot treure, ja que es redueixen els temps d'execució, el nombre d'instruccions usades i la capacitat de processament requerides.

8 Referències

1. YouTube,
   https://www.youtube.com/

2. Wikipedia: YouTube,
   https://ca.wikipedia.org/wiki/YouTube

3. Random YouTube Videos,
   https://randomyoutube.net/

4. sitelabs: Qué es el Web scraping?,
   https://sitelabs.es/web-scraping-introduccion-y-herramientas/

5. Wikipedia: Modelo Lineal Generalizado,
   https://es.wikipedia.org/wiki/Modelo_lineal_generalizado

6. Apunts i diapositives de l'assignatura Mineria de Dades

7. Apunts i diapositives de l'assignatura Eines Informàtiques

9 Metadata

Títol del Treball
Web Scraping de Vídeos de YouTube i Realització d'un Model Predictiu per a la seva Visualització

Autor
Arnau Rovira Chassignet

Tutor
Albert Ruiz Cirera

9.1 Resums

Resum
Aquest treball ha consistit en l'extracció, mitjançant el mètode del "Web Scraping", de la informació corresponent als vídeos de l'historial del meu compte de YouTube. D'altra banda també comptava amb una altra base de vídeos aleatòris de la plataforma. Posteriorment he depurat, estudiat i transformat les dades per tal de poder realitzar un seguit de models predictius amb aquestes. A continuació, mitjançant una base de dades d'entrenament i una de validació, i contrastant un seguit d'estadístics, he optat per un model final com el millor predictor. Finalment les conclusions han estat que les dades de l'historial no eren representatives de l'univers dels vídeos de YouTube. Això és així ja que les variables explicatives prenien distribucions molt diferents segons si el vídeo provenia de l'historial o de la mostra aleatòria, i per tant tots els models proposats tenien un percentatge d'encert extremadament alt.

Resumen
Este trabajo ha consistido en la extración, mediante el método del "Web Scraping", de la información correspondiente a los vídeos del historial de mi cuenta de YouTube. Por otro lado también he podido contar con una muestra de vídeos aleatorios de la plataforma. Posteriormente ha depurado, estudiado y transformado los datos para poder realizar una série de modelos predictivos. A continuación, mediante una base de datos de entrenamiento y otra de validación, y contrastando ciertos estadísticos, he optado por un modelo final como el mejor predictor. Finalmente las conclusiones extraídas han sido que los datos correspondientes a mi historial no eran representativos del universo de vídeos de Youtube. Esto es así puesto que las variables explicativas tomaban distribuciones muy diferentes según si el vídeo provenia del historial o de la muestra aleatoria, y por tanto todos los modelos propuestos tenian un alto porcentaje de acierto.

Abstract
This work consisted in the extraction, by means of the "Web Scraping" method, of the information corresponding to the videos of the history of my YouTube account. On the other hand I have also been able to have a sample of random videos of the platform. Subsequently, he has refined, studied and transformed the data in order to carry out a series of predictive models. Then, through a database of training and validation, and contrasting certain statistics, I opted for a final model as the best predictor. Finally the conclusions drawn have been that the data corresponding to my history were not representative of the universe of YouTube videos. This is so since the explanatory variables took very different distributions depending on whether the video came from the history or from the random sample, and therefore all the proposed models had a high percentage of success.

Paraules Clau
Web Scraping, Bash, R, Model Predictiu

A Codi R

Codi R necessari per al tractament de les dades i la creació de models.

```
mmss_to_ss <- function(string)
{
  mmss <- strsplit(string, ":", T)
  mm <- as.numeric(mmss[[1]][1])
  ss <- as.numeric(mmss[[1]][2])
  return (mm * 60 + ss)
}

transformador <- function(cadena)
{
  noobs <- length(cadena)
  cadena2 <- numeric(noobs)
  for(i in 1:length(cadena))
  {
    if(grepl("M", cadena[i], fixed = T))
    {
      cadena2[i] <- as.numeric(gsub("M", "", cadena[i]))*1000000
    }
    else
    {
      if(grepl("K", cadena[i], fixed = T))
      {
        cadena2[i] <- as.numeric(gsub("K", "", cadena[i]))*1000
      }
      else
      {
        cadena2[i] <- as.numeric(cadena[i])
      }
    }
  }
  cadena2
}

anycat <- function(cadena)
{
  noobs <- length(cadena)
  cadena2 <- numeric(noobs)
  for(i in 1:length(cadena))
  {
    if(cadena[i] <= 2008)
    {
      cadena2[i] <- as.character("2005-2008")
    }
    else
    {
      if(cadena[i] <= 2011)
      {
        cadena2[i] <- as.character("2009-2011")
```

```
      }
      else
      {
        if(cadena[i] <= 2014)
        {
          cadena2[i] <- as.character("2012-2014")
        }
        else
        {
          cadena2[i] <- as.character("2015-2017")
        }
      }
    }
  }
  cadena2
}


dades <- data.frame(read.csv("fitxer.csv", header = T, sep="\t"))
nobsini <- dim(dades)[1]
dades2 <- dades[complete.cases(dades), ]
dades2 <- unique(dades2)
head(dades2)
summary(dades2)
dades2 <- dades2[dades2$Subscriptors != "", ]
nobs <- dim(dades2)[1]; nobs


dades2$Durada <- as.character(dades2$Durada)
dades2$Segons <- as.numeric(nobs)
for(i in 1:nobs)
{
  dades2$Segons[i] <- mmss_to_ss(dades2$Durada[i])
}
summary(dades2)

dades2$Visites <- as.numeric(as.character(dades2$Visites))
dades2$Agrada <- as.numeric(as.character(dades2$Agrada))
dades2$NoAgrada <- as.numeric(as.character(dades2$NoAgrada))
dades2$Visualitzat <- as.factor(dades2$Visualitzat)

dades2$Subscriptors <- transformador(dades2$Subscriptors)
dades2$DataCat <- anycat(as.numeric(substr(dades2$Data, 1, 4)))
dades2$DataCat <- as.factor(dades2$DataCat)

head(dades2)
summary(dades2)
dades2 <- na.omit(dades2)

summary(dades2$Genere)
dades2$Genere <- gsub("&amp;", "&", dades2$Genere)
```

```r
dades2$Genere <- as.factor(dades2$Genere)


taulaGenere <- as.table(table(dades2$Genere)[table(dades2$Genere)!=0])
xtabGenere <- xtable(taulaGenere, caption="Taula de freqüències de la variable Genere."
                     , label="tab:Genere")
names(xtabGenere) <- c('Gènere')

taulaVisualitzat <- as.table(summary(dades2$Visualitzat))
xtabVisualitzat <- xtable(taulaVisualitzat, caption="Taula de freqüències de la variable
                          Visualitzat.", label="tab:Visualitzat")
names(xtabVisualitzat) <- c('Visualitzat')

taulaDataCat <- as.table(summary(dades2$DataCat))
xtabDataCat <- xtable(taulaDataCat, caption="Taula de freqüències de la variable DataCat
                      .", label="tab:DataCat")
names(xtabDataCat) <- c('DataCat')


taulaGenereprop <- round(table(dades2$Genere)/sum(table(dades2$Genere))*100, 2)
taulaGenereprop <- taulaGenereprop[taulaGenereprop!=0]; taulaGenereprop

dades3 <- subset(dades2, dades2$Genere!="Trailers")
dades3<- subset(dades3, dades3$Genere!="Shows")
names(dades3)
dades3 <- dades3[-c(5,7)]
dades3$Visualitzat <- as.factor(dades3$Visualitzat)
head(dades3)
summary(dades3)
nobs <- dim(dades3)[1]; nobs


resdades3 <- xtable(summary(dades3))

dcai_Segons <- boxplot(dades3$Segons)
dades3$logSegons <- log(dades3$Segons+1)
dcai_logSegons <- boxplot(dades3$logSegons)
shapiro.test(dades3$logSegons) #no normal

dcai_Visites <- boxplot(dades3$Visites)
dades3$logVisites <- log(dades3$Visites)
dcai_logVisites <- boxplot(dades3$logVisites)
shapiro.test(dades3$logVisites) #no normal

dcai_Agrada <- boxplot(dades3$Agrada)
dades3$logAgrada <- log(dades3$Agrada+1)
dcai_logAgrada <- boxplot(dades3$logAgrada)
shapiro.test(dades3$logAgrada) #no normal

dcai_NoAgrada <- boxplot(dades3$NoAgrada)
dades3$logNoAgrada <- log(dades3$NoAgrada+1)
```

```
dcai_logNoAgrada <- boxplot(dades3$logNoAgrada)
shapiro.test(dades3$logNoAgrada) #no normal

dcai_Subscriptors <- boxplot(dades3$Subscriptors)
dades3$logSubscriptors <- log(dades3$Subscriptors+1)
dcai_logSubscriptors <- boxplot(dades3$logSubscriptors)
shapiro.test(dades3$logSubscriptors) #no normal

tauVisualitzat <- table(dades3$Visualitzat)
tauGenere <- table(dades3$Genere)
tauDataCat <- table(dades3$DataCat)

xtVisualitzat <- xtable(tauVisualitzat)
names(xtVisualitzat) <- c('Visualitzat')
xtGenere <- xtable(tauGenere, caption="Distribució definitiva de la variable Genere."
                   , label="tab:Genere2")
names(xtGenere) <- c('Gènere')
xtDataCat <- xtable(tauDataCat)
names(xtDataCat) <- c('Data')


nobs <- dim(dades3)[1]; nobs

correlacions <- xtable(as.table(cor(dades3[,c(9:13)])), caption="Correlacions entre les
                       variables numèriques.", label="tab:correlacions")
grafcorrelacions <- pairs(dades3[,c(9:13)])

videosvistos <- subset(dades3, dades3$Visualitzat==1)
videosNOvistos <- subset(dades3, dades3$Visualitzat==0)

resvistos <- xtable(summary(videosvistos))
resNOvistos <- xtable(summary(videosNOvistos))

boxplot(videosvistos$logSegons)
boxplot(videosNOvistos$logSegons)
dcaicomp_logSegons <- boxplot(videosvistos$logSegons, videosNOvistos$logSegons,
                              names=c("Visualitzat", "No Visualitzat"));
                              title("logSegons")

boxplot(videosvistos$logAgrada)
boxplot(videosNOvistos$logAgrada)
dcaicomp_logAgrada <- boxplot(videosvistos$logAgrada, videosNOvistos$logAgrada,
                              names=c("Visualitzat", "No Visualitzat"));
                              title("logAgrada")

boxplot(videosvistos$logNoAgrada)
boxplot(videosNOvistos$logNoAgrada)
dcaicomp_logNoAgrada <- boxplot(videosvistos$logNoAgrada, videosNOvistos$logNoAgrada,
                                names=c("Visualitzat", "No Visualitzat"));
                                title("logNoAgrada")
```

```
boxplot(videosvistos$logVisites)
boxplot(videosNOvistos$logVisites)
dcaicomp_logVisites <- boxplot(videosvistos$logVisites, videosNOvistos$logVisites,
                               names=c("Visualitzat", "No Visualitzat"));
                               title("logVisites")

boxplot(videosvistos$logSubscriptors)
boxplot(videosNOvistos$logSubscriptors)
dcaicomp_logSubscriptors <- boxplot(videosvistos$logSubscriptors,
                                    videosNOvistos$logSubscriptors,
                                    names=c("Visualitzat", "No Visualitzat"));
                                    title("logSubscriptors")

table(dades3$Visualitzat, dades3$Genere)
vistosGenere <- round(table(videosvistos$Genere)/sum(table(videosvistos$Genere))*100, 2)
vistosGenere <- vistosGenere[vistosGenere!=0]; vistosGenere
NOvistosGenere <- round(table(videosNOvistos$Genere)/sum(table(videosNOvistos$Genere
                                                               ))*100, 2)
NOvistosGenere <- NOvistosGenere[NOvistosGenere!=0]; NOvistosGenere

table(dades3$Visualitzat, dades3$DataCat)
vistosDataCat <- round(table(videosvistos$DataCat)/sum(table(videosvistos$DataCat))*100,
                       2)
NOvistosDataCat <- round(table(videosNOvistos$DataCat)/sum(table(videosNOvistos$DataCat
                                                                 ))*100, 2)



names(dades3)
dades4 <- dades3[,c(5,6,8:13)]
summary(dades4)
nobs <- dim(dades4)[1]

mostra <- sample(1:nobs, 1100)
dadesvali <- dades4[mostra,]
dadestest <- dades4[-mostra,]


#Model Lineal Generalitzat

mod11 <- glm(Visualitzat ~ ., data=dadestest, family="binomial")
summary(mod11)
pred11 <- predict(mod11, newdata=dadesvali, type="response")
taula1 <- table(real=dadesvali$Visualitzat, predit=round(pred11, 0)); taula1
PEC11 <- 1-sum(diag(taula1))/sum(taula1); PEC11
ROC11 <- roc(dadesvali$Visualitzat, pred11, plot = T)
AUC11 <- roc(dadesvali$Visualitzat, pred11, plot = T)$auc; AUC11
EQM11 <- mean((pred11-(as.numeric(dadesvali$Visualitzat)-1))^2); EQM11


#Arbre de Regressió 1
```

```r
mod21 <- rpart(Visualitzat~., data=dadestest)
plot(mod21); text(mod21)
pred21 <- predict(mod21, newdata = dadesvali, type="prob")[,2]
taula2 <- table(real=dadesvali$Visualitzat,predit=round(pred21, 0)); taula2
PEC21<- 1-sum(diag(taula2))/sum(taula2); PEC21
ROC21 <- roc(dadesvali$Visualitzat, pred21, plot = T)
AUC21 <- roc(dadesvali$Visualitzat, pred21, plot = T)$auc; AUC21
EQM21 <- mean((pred21-(as.numeric(dadesvali$Visualitzat)-1))^2); EQM21


#Arbre de Regressió 2

mod22 <- rpart(Visualitzat~., data=dadestest, cp = 0.001)
plot(mod22); text(mod22)
pred22 <- predict(mod22, newdata = dadesvali, type="prob")[,2]
taula3 <- table(real=dadesvali$Visualitzat,predit=round(pred22, 0)); taula3
PEC22 <- 1-sum(diag(taula3))/sum(taula3); PEC22
ROC22 <- roc(dadesvali$Visualitzat, pred22, plot = T)
AUC22 <- roc(dadesvali$Visualitzat, pred22, plot = T)$auc; AUC22
EQM22 <- mean((pred22-(as.numeric(dadesvali$Visualitzat)-1))^2); EQM22


#Xarxa Neuronal 1

aucaux <- 0
modaux <- NA
predeaux <- NA
for (i in 1:3)
{
  modaux <- nnet(Visualitzat~., data=dadestest, size=4, maxit=500, linout=F)

  predaux <- as.vector(predict(modaux, newdata=dadesvali, type="raw"))

  if(roc(dadesvali$Visualitzat, predaux)$auc > aucaux)
  {
    mod31 <- modaux
    aucaux <- roc(dadesvali$Visualitzat, predaux)$auc
    pred31 <- predaux
  }
  print(i)
}
taula4 <- table(real = dadesvali$Visualitzat, predit=round(pred31, 0)); taula4
PEC31 <- 1-sum(diag(taula4))/sum(taula4); PEC31
ROC31 <- roc(dadesvali$Visualitzat, pred31, plot = T)
AUC31 <- roc(dadesvali$Visualitzat, pred31, plot = T)$auc; AUC31
EQM31 <- mean((pred31-(as.numeric(dadesvali$Visualitzat)-1))^2); EQM31


#Xarxa Neuronal 2

aucaux <- 0
```

```
modaux <- NA
predeaux <- NA
for (i in 1:3)
{
  modaux <- nnet(Visualitzat~., data=dadestest, size=6, maxit=1000, linout=F, skip=T)

  predaux <- as.vector(predict(modaux, newdata=dadesvali, type="raw"))

  if(roc(dadesvali$Visualitzat, predaux)$auc > aucaux)
  {
    mod32 <- modaux
    aucaux <- roc(dadesvali$Visualitzat, predaux)$auc
    pred32 <- predaux
  }
  print(i)
}
taula5 <- table(real = dadesvali$Visualitzat, predit=round(pred32, 0)); taula4
PEC32 <- 1-sum(diag(taula5))/sum(taula5); PEC32
ROC32 <- roc(dadesvali$Visualitzat, pred32, plot = T)
AUC32 <- roc(dadesvali$Visualitzat, pred32, plot = T)$auc; AUC32
EQM32 <- mean((pred32-(as.numeric(dadesvali$Visualitzat)-1))^2); EQM32


#Random Forest 1

dadestestaux <- dadestest
dadesvaliaux <- dadesvali
mod41 <- randomForest(Visualitzat~., data=dadestestaux, ntree=500, nodesize=1)
pred41 <- predict(mod41, newdata=dadesvaliaux, type="prob")[, 2]
taula7 <- table(real = dadesvaliaux$Visualitzat, predit=round(pred41, 0)); taula7
PEC41 <- 1-sum(diag(taula7))/sum(taula7); PEC41
ROC41 <- roc(dadesvaliaux$Visualitzat, pred41, plot = T)
AUC41 <- roc(dadesvaliaux$Visualitzat, pred41, plot = T)$auc; AUC41
EQM41 <- mean((pred41-(as.numeric(dadesvaliaux$Visualitzat)-1))^2); EQM41


#Random Forest 2

dadestestaux <- dadestest
dadesvaliaux <- dadesvali
mod42 <- randomForest(Visualitzat~., data=dadestestaux, ntree=800, nodesize=1, mtry=2)
pred42 <- predict(mod42, newdata=dadesvaliaux, type="prob")[, 2]
taula7 <- table(real = dadesvaliaux$Visualitzat, predit=round(pred42, 0)); taula7
PEC42 <- 1-sum(diag(taula7))/sum(taula7); PEC42
ROC42 <- roc(dadesvaliaux$Visualitzat, pred42, plot = T)
AUC42 <- roc(dadesvaliaux$Visualitzat, pred42, plot = T)$auc; AUC42
EQM42 <- mean((pred42-(as.numeric(dadesvaliaux$Visualitzat)-1))^2); EQM42
```

B Taules

	Gènere
Autos & Vehicles	53
Comedy	135
Education	61
Entertainment	555
Film & Animation	114
Gaming	1089
Howto & Style	51
Music	1281
News & Politics	77
Nonprofits & Activism	13
People & Blogs	601
Pets & Animals	18
Science & Technology	93
Shows	5
Sports	162
Trailers	1
Travel & Events	34

Taula 7: Taula de freqüències de la variable Genere.

	DataCat
2005-2008	76
2009-2011	336
2012-2014	652
2015-2017	3279

Taula 8: Taula de freqüències de la variable DataCat .

	Visualitzat
0	837
1	3506

Taula 9: Taula de freqüències de la variable Visualitzat.

	Visualitzat	No Visualitzat
Autos & Vehicles	0.71	3.35
Comedy	3.20	2.75
Education	0.80	3.94
Entertainment	14.34	6.33
Film & Animation	2.66	2.51
Gaming	27.83	13.74
Howto & Style	1.06	1.67
Music	34.74	7.77
News & Politics	1.34	3.58
Nonprofits & Activism	0.14	0.96
People & Blogs	6.31	45.40
Pets & Animals	0.31	0.84
Science & Technology	2.11	2.27
Sports	3.80	3.46
Travel & Events	0.63	1.43

Taula 10: Taula de freqüències relatives de la variable Genere respecte el valor de la variable Visualitzat.

	Visualitzat	No Visualitzat
2005-2008	2.09	0.36
2009-2011	9.14	1.91
2012-2014	16.20	9.44
2015-2017	72.57	88.29

Taula 11: Taula de freqüències relatives de la variable DataCat respecte el valor de la variable Visualitzat.

C Gràfics

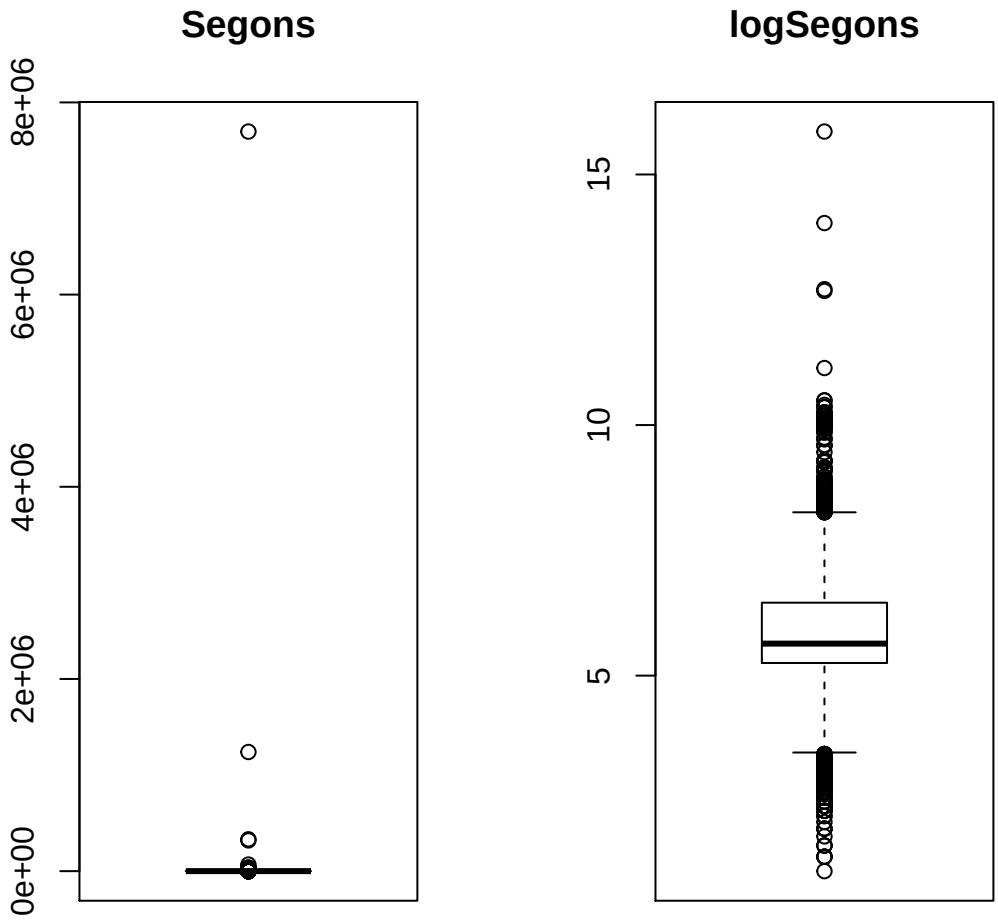

Figura 4: Diagrames de caixa per a les variables Segons i logSegons.

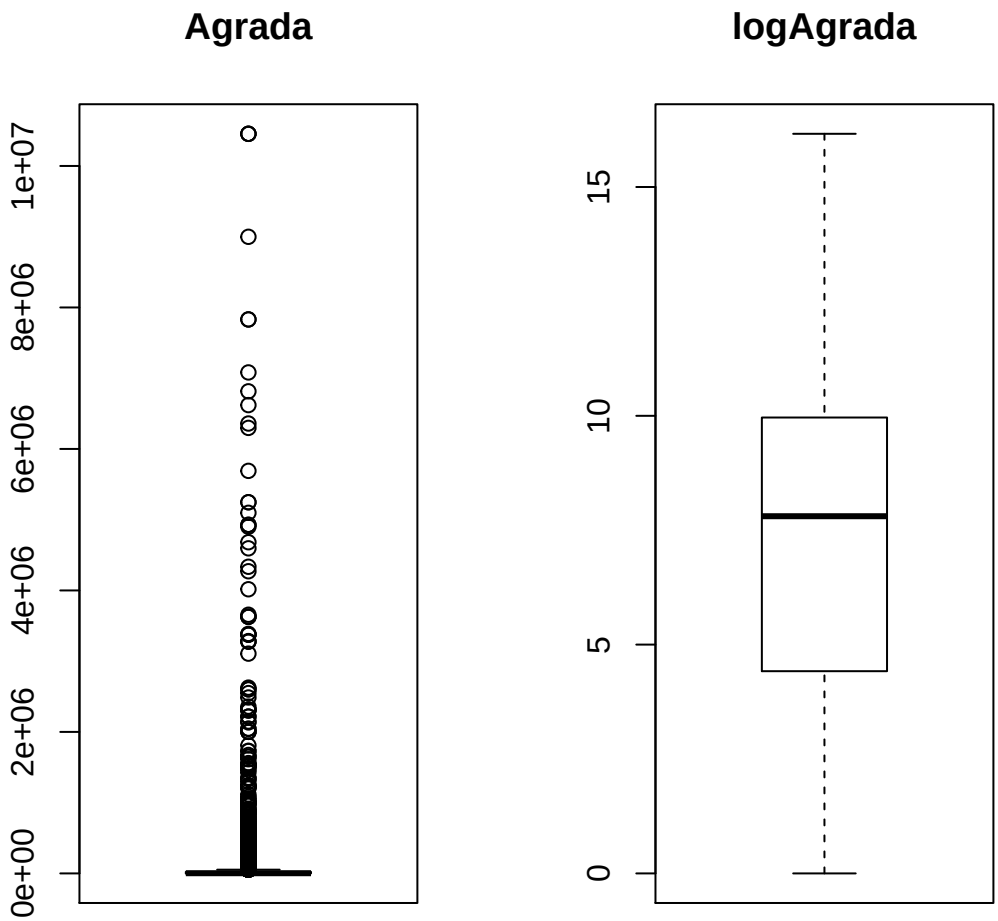

Figura 5: Diagrames de caixa per a les variables Agrada i logAgrada.

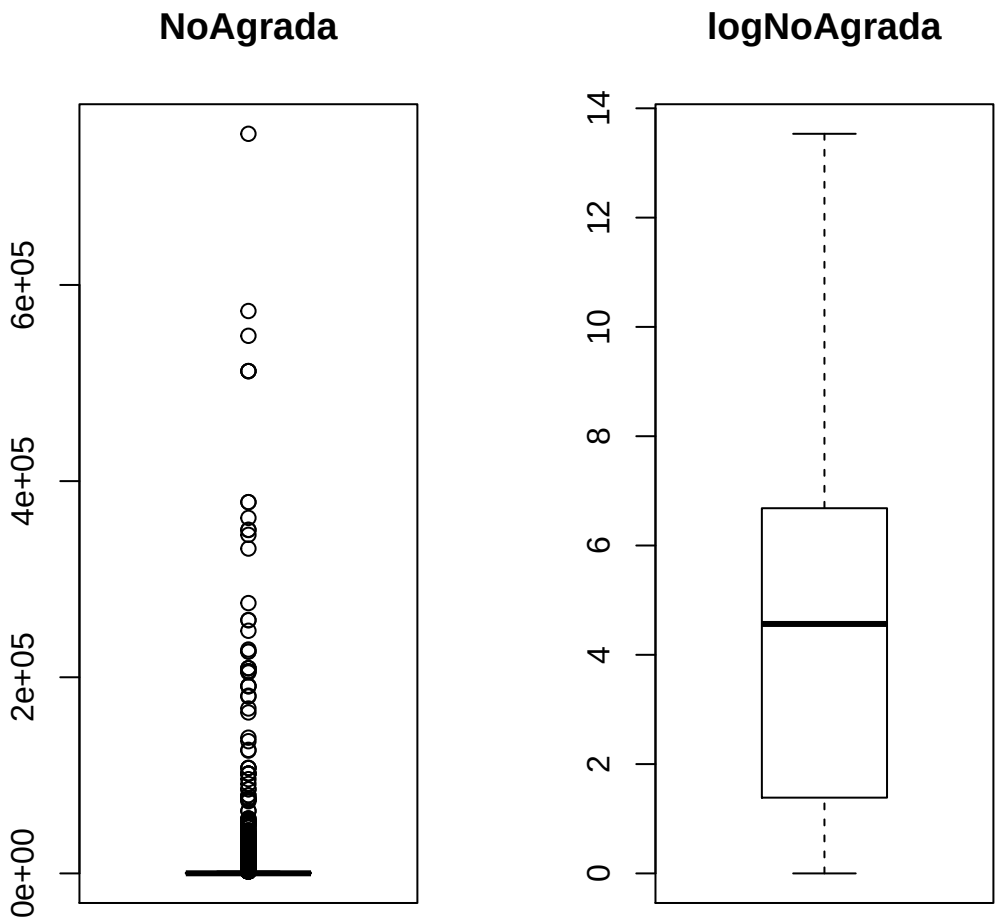

Figura 6: Diagrames de caixa per a les variables NoAgrada i logNoAgrada.

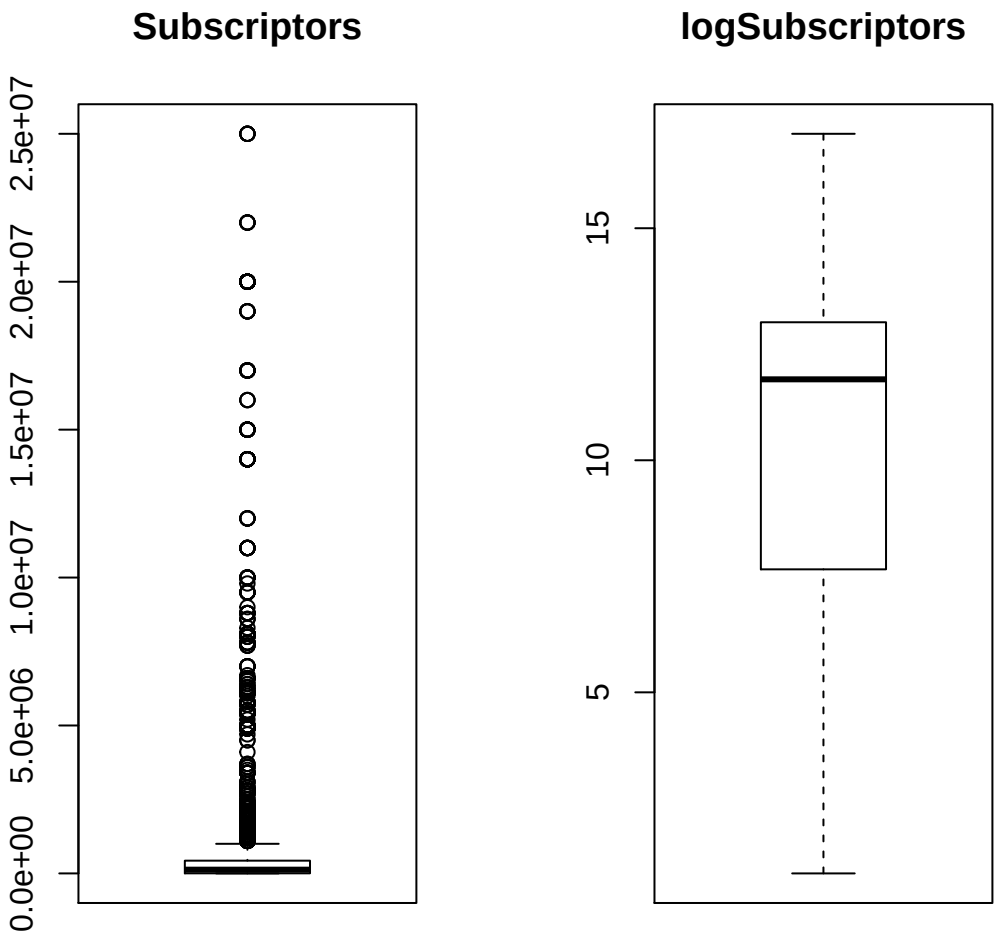

Figura 7: Diagrames de caixa per a les variables Subscriptors i logSubscriptors.

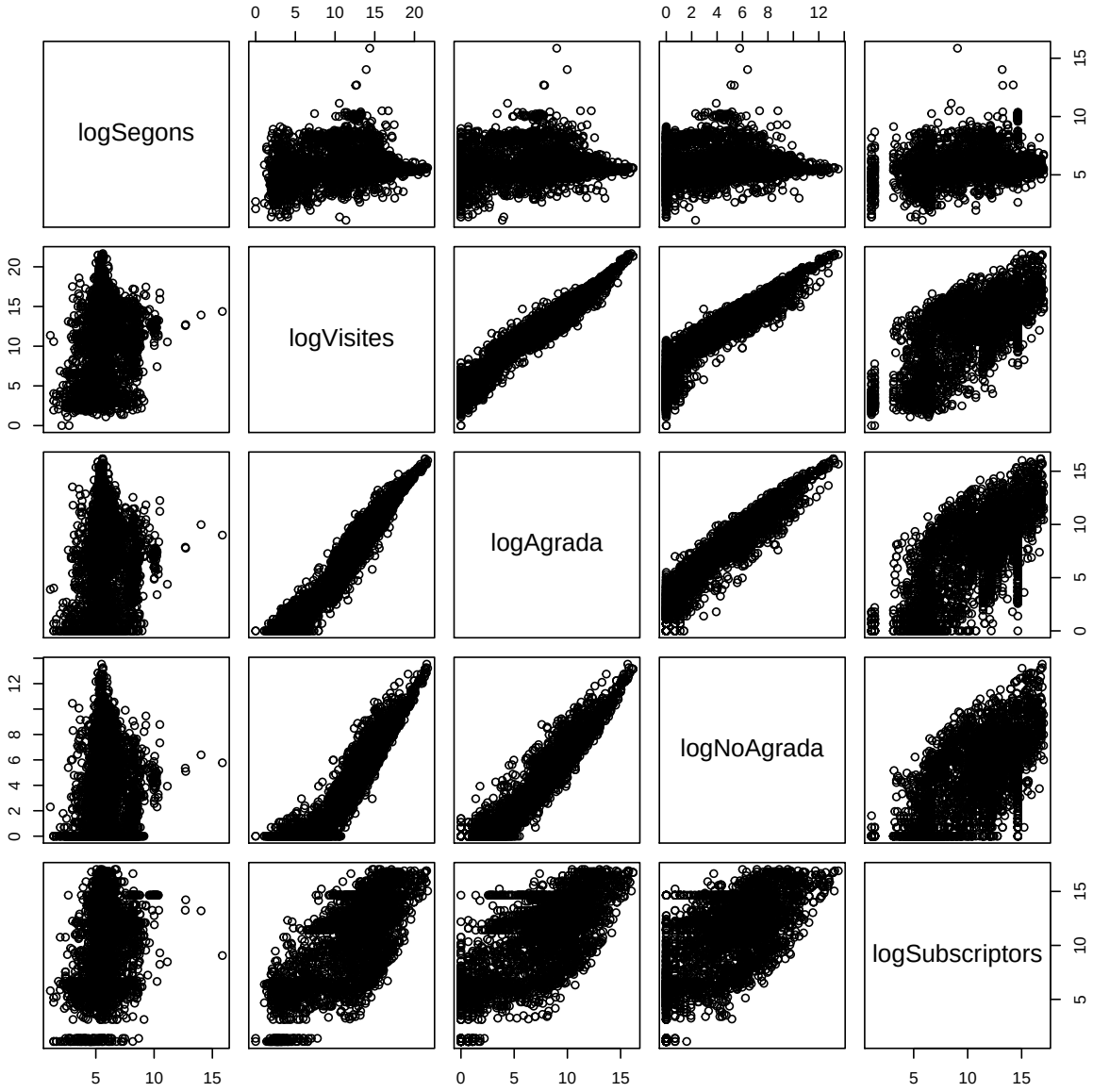

Figura 8: Núvols de punts segons les varibles numèriques.

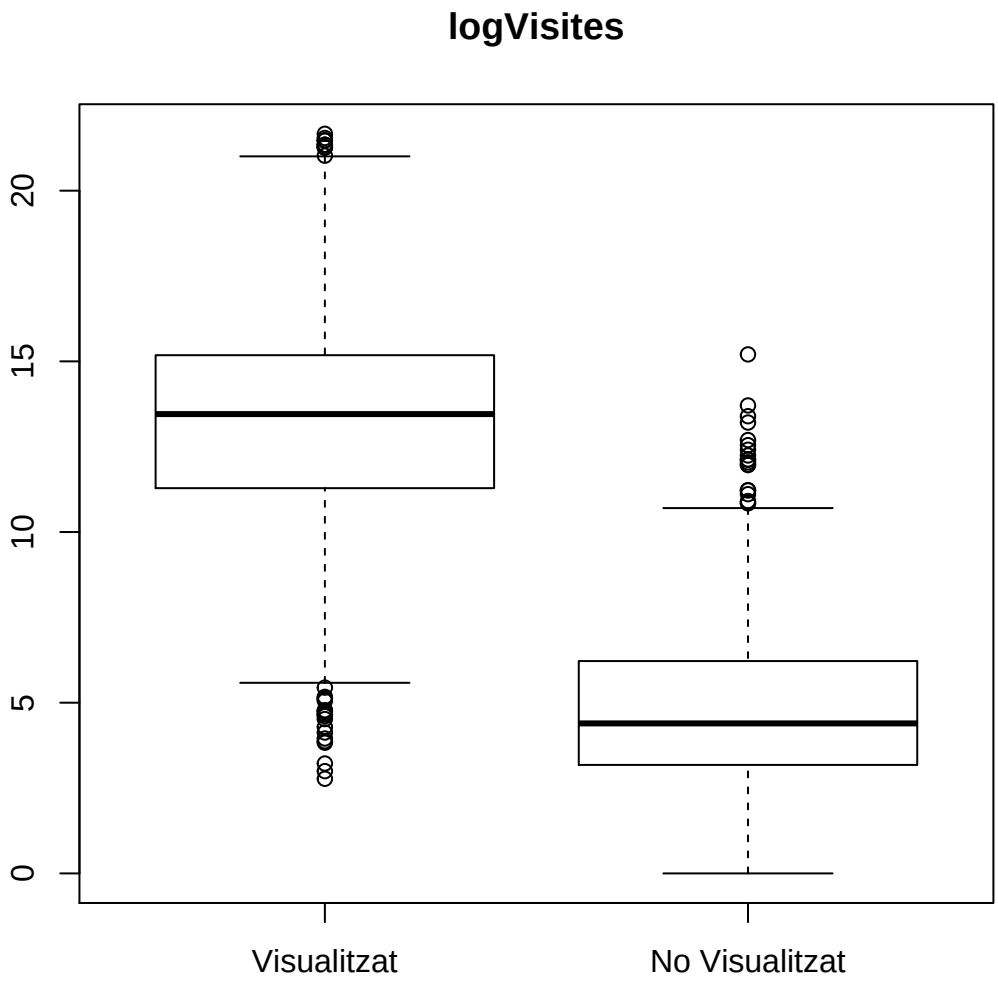

Figura 9: Diagrames de caixa per a la variable logVisites segons el valor de la variable Visualitzat.

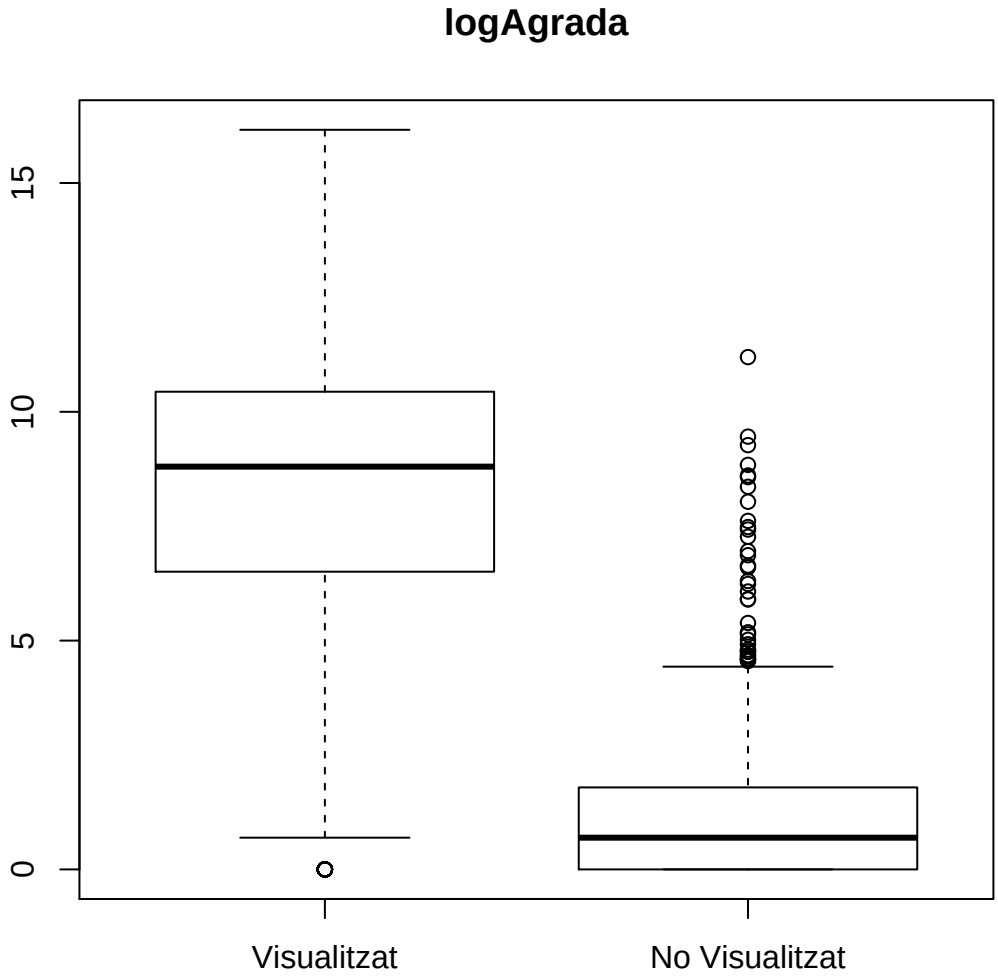

Figura 10: Diagrames de caixa per a la variable logAgrada segons el valor de la variable Visualitzat.

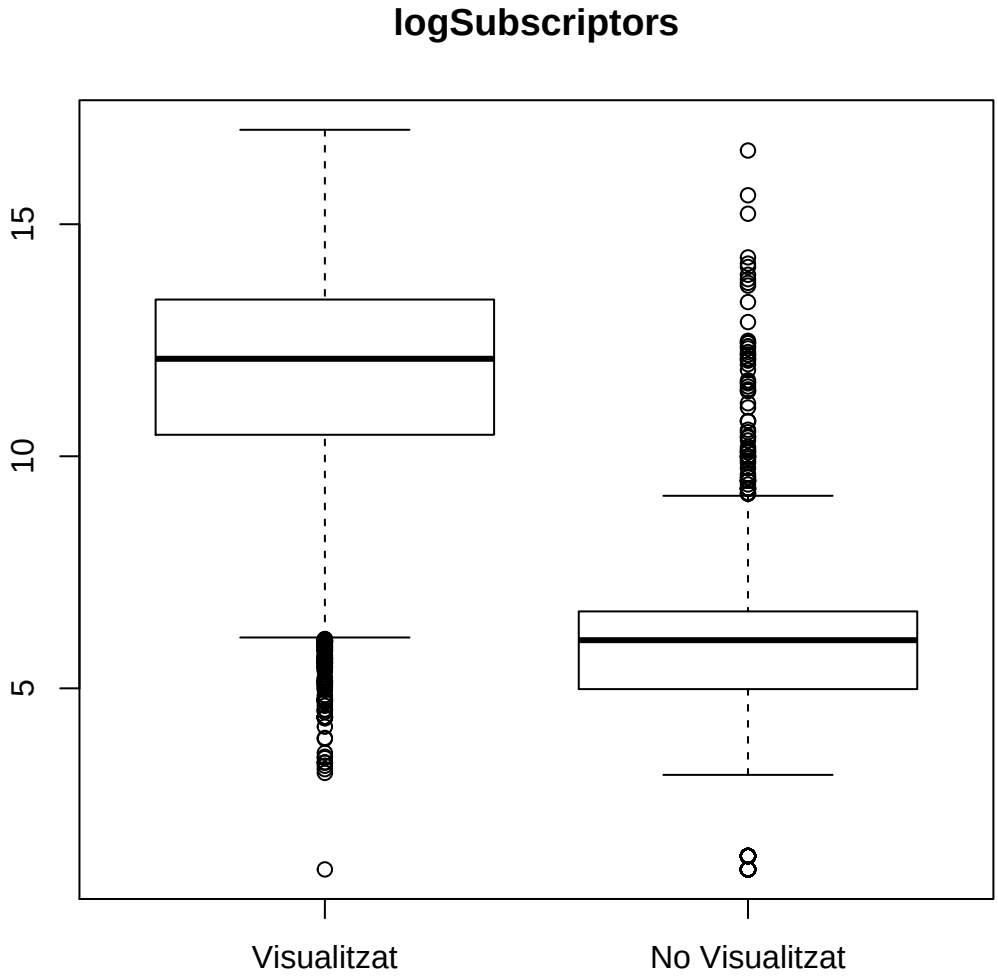

Figura 11: Diagrames de caixa per a la variable logSubscriptors segons el valor de la variable Visualitzat.

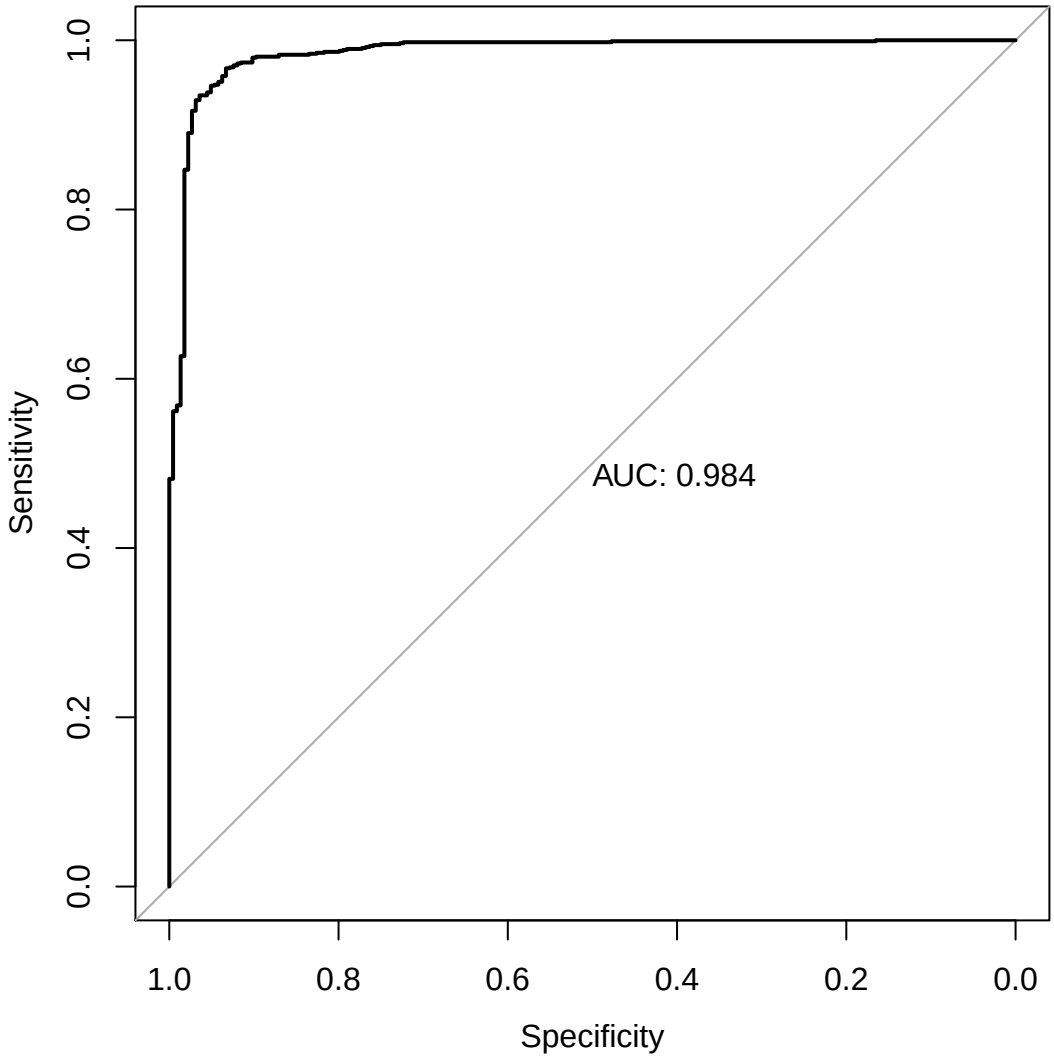

Figura 12: Corba ROC del model lineal generalitzat.

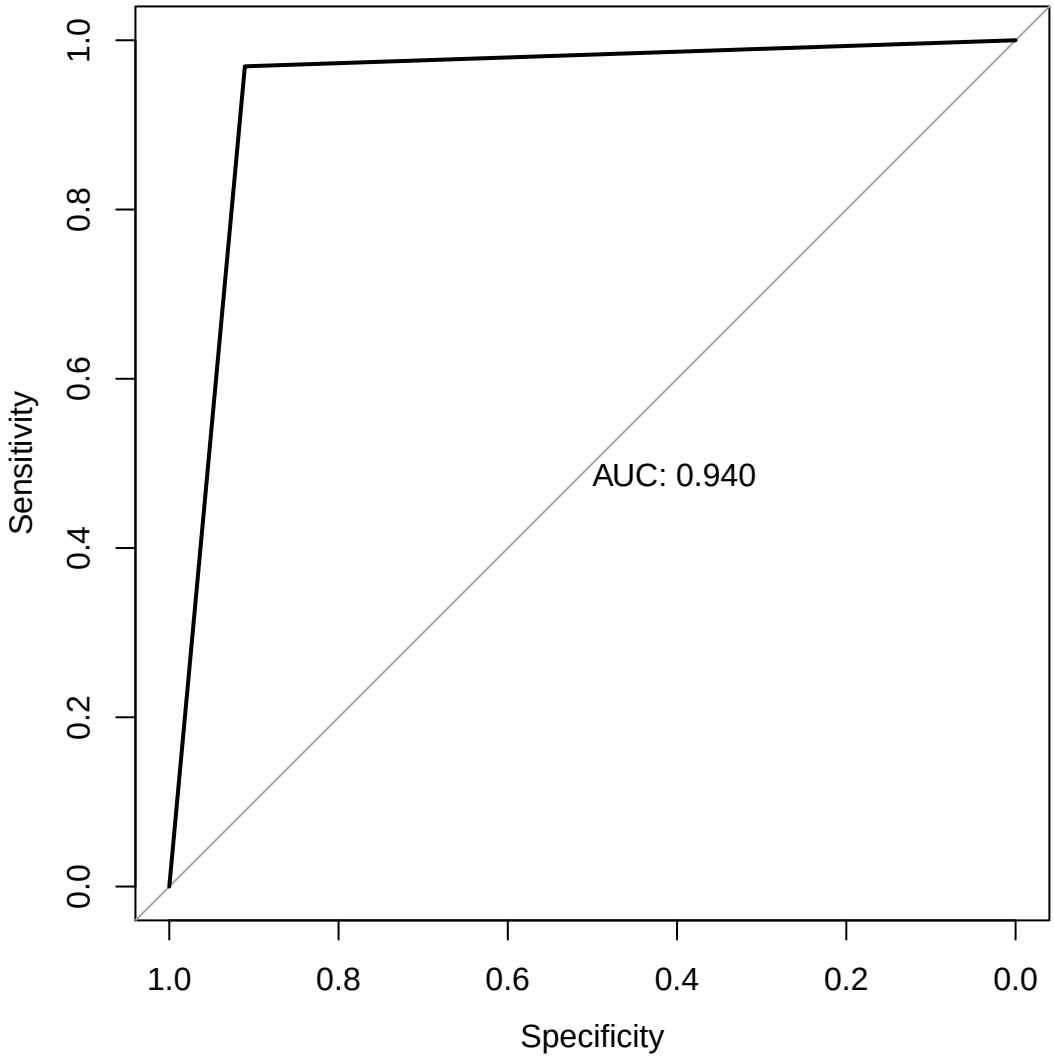

Figura 13: Corba ROC de l'arbre de classificació inicial.

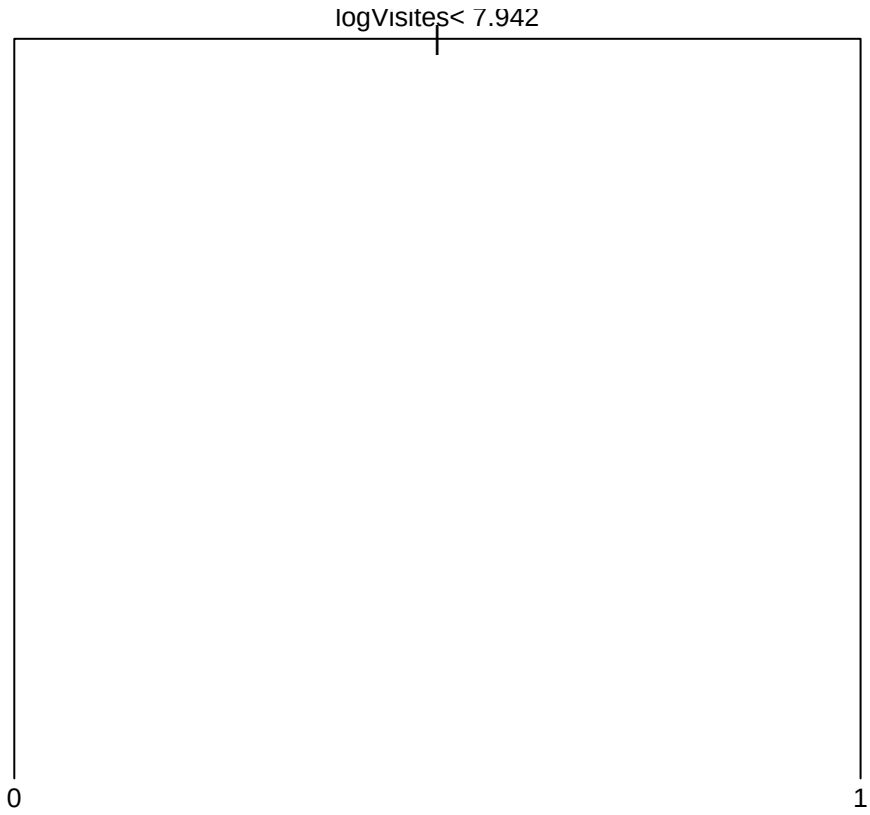

Figura 14: Estructura de l'arbre de classificació inicial.

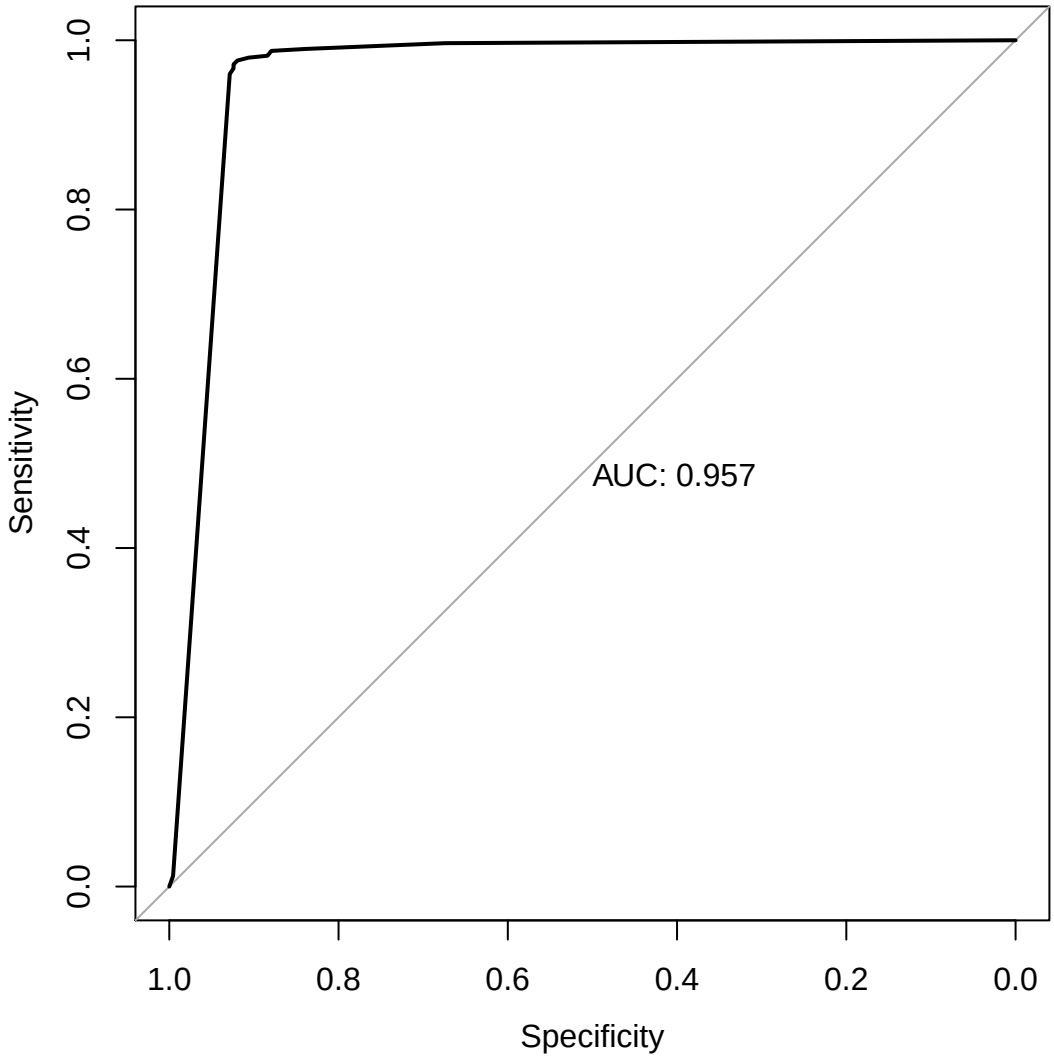

Figura 15: Corba ROC de l'arbre de classificació definitiu.

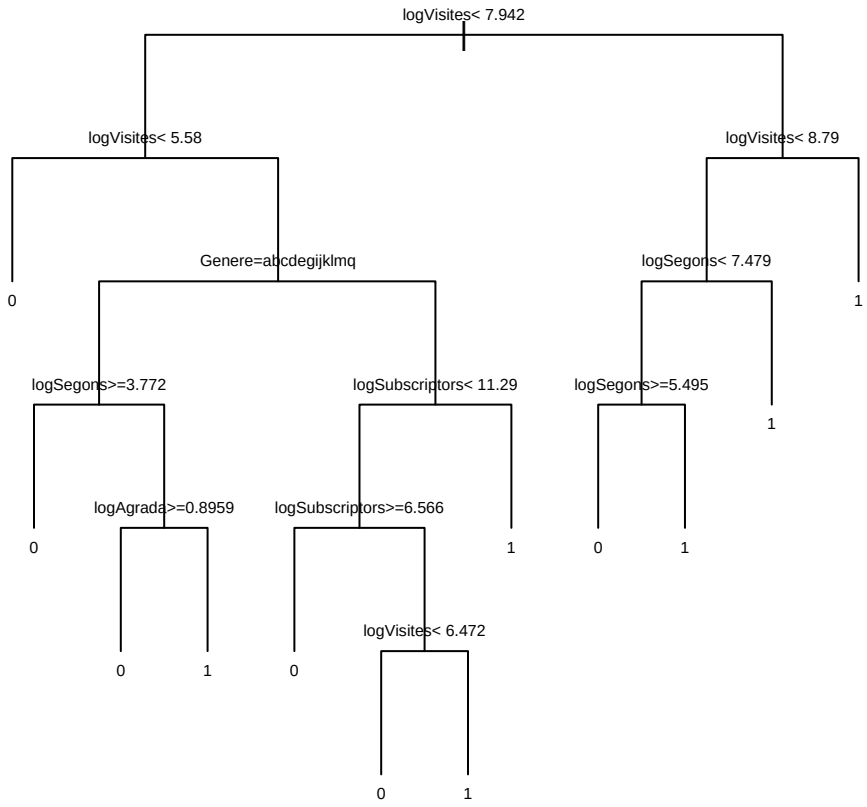

Figura 16: Estructura de l'arbre de classificació definitiu.

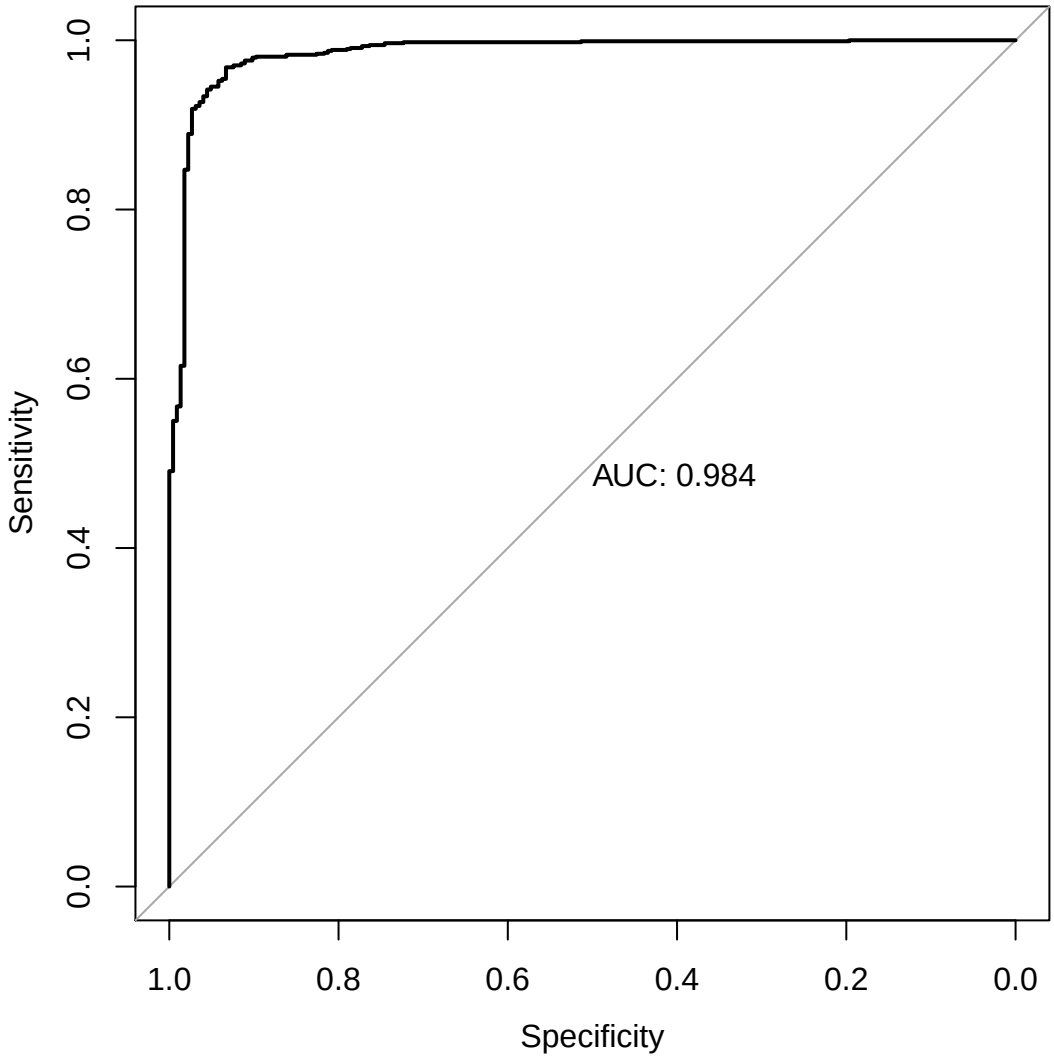

Figura 17: Corba ROC de la xarxa neuronal inicial.

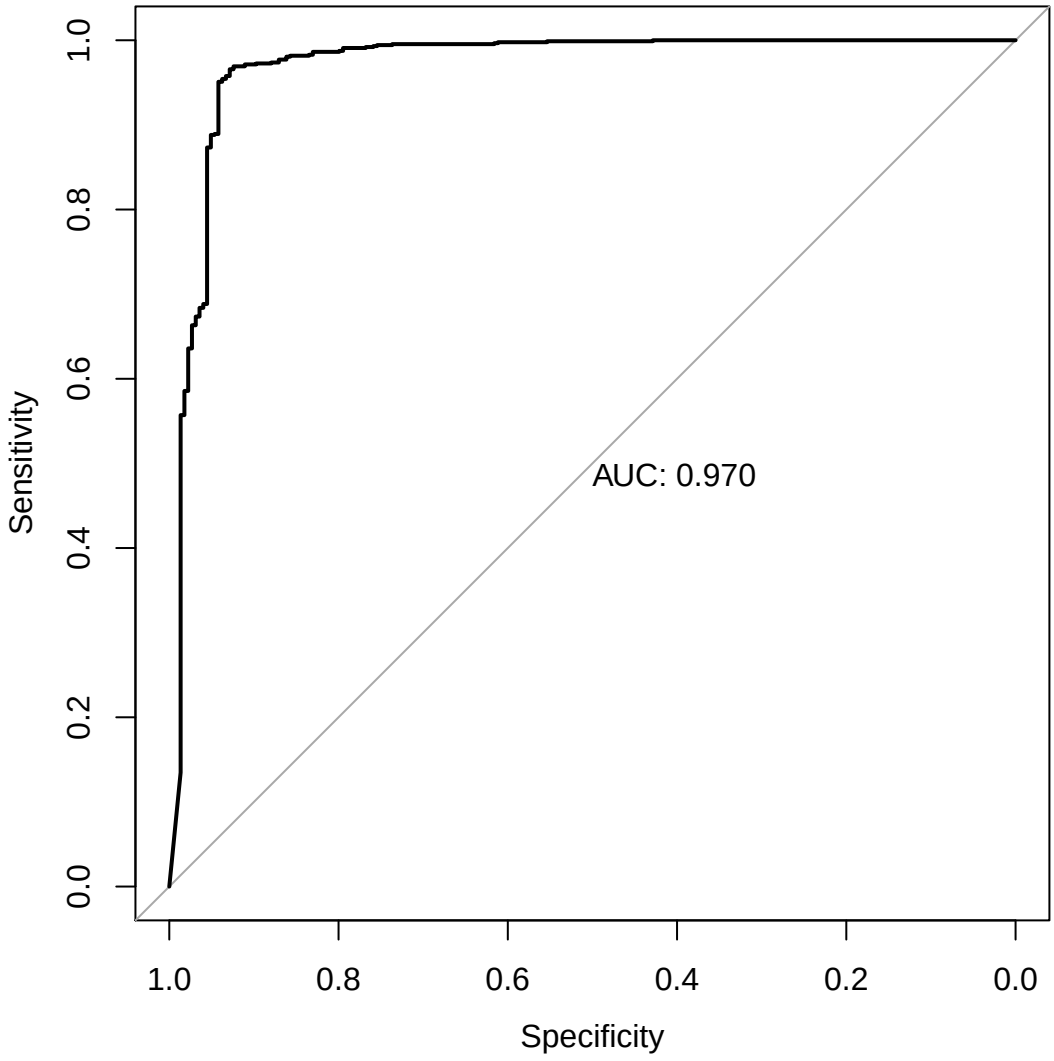

Figura 18: Corba ROC de la xarxa neuronal definitiva.

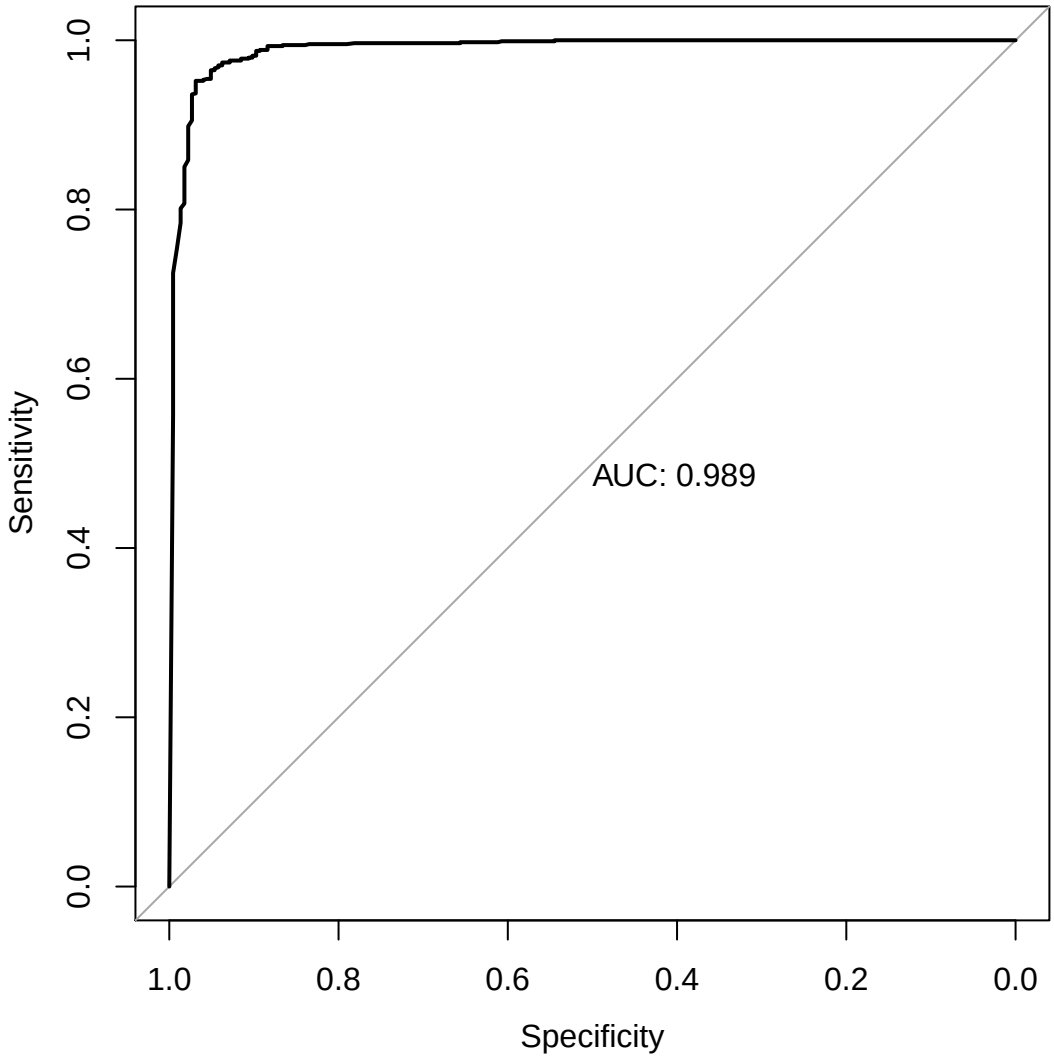

Figura 19: Corba ROC del random forest inicial.